Calhoun: The NPS Institutional Archive

DSpace Repository

Theses and Dissertations 1. Thesis and Dissertation Collection, all items

1985-09

Computer aided instruction design issues: the development of a portable microcomputer-based CAI for statistics instruction

Henderson, Randall L.

http://hdl.handle.net/10945/21207

This publication is a work of the U.S. Government as defined in Title 17, United States Code, Section 101. Copyright protection is not available for this work in the United States.

Downloaded from NPS Archive: Calhoun

Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

http://www.nps.edu/library

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA 93943

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

COMPUTER AIDED INSTRUCTION DESIGN ISSUES: THE DEVELOPMENT OF A PORTABLE MICROCOMPUTER-BASED CAI FOR STATISTICS INSTRUCTION

by

Randall L. Henderson

September 1985

Co-Advisors: Tung Bui
F.Marchman Perry

Approved for public release; distribution is unlimited

T224196

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Computer Aided Instruction Design Issues: The Development of a Portable Microcomputer-Based CAI for Statistics Instruction		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis September 1985
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Randall L. Henderson		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93943		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93943		12. REPORT DATE September 1985
		13. NUMBER OF PAGES 292
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution is unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

CAI, Computer Assisted Instruction, Statistical Programs
Computers and Education

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The primary focus of this thesis is the development of a portable microcomputer-based software package to aid in the instruction of a graduate level introductory statistics course. In order to accomplish this objective, an exploration of various conventional Computer Assisted Instruction (CAI) techniques is necessary to determine which methods are readily adaptable for use with portable microcomputers. Many of the design issues normally associated with the development of CAI packages (Continued)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S N 0102- LF- 014- 6601

ABSTRACT (Continued)

are exasperated by the current physical limitations of portable microcomputers. This thesis will discuss general CAI design issues and related compromises and trade-offs needed to successfully implement some of these designs in a portable microcomputer-based system. A systematic methodology for designing simple paired-associate learning (Drill and Practice) software is presented, which incorporates many of the lessons learned in the development of the included statistics software package.

Approved for public release, distribution unlimited

Computer Aided Instruction Design Issues: The Development of a Portable Microcomputer-Based CAI for Statistics Instruction

by

Randall L. Henderson
Lieutenant, United States Navy
B.S., University of South Carolina, 1976

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS

from the

NAVAL POSTGRADUATE SCHOOL
September 1985

Thesis
[illegible]
c.1

ABSTRACT

The primary focus of this thesis is the development of a portable microcomputer-based software package to aid in the instruction of a graduate level introductory statistics course. In order to accomplish this objective, an exploration of various conventional Computer Assisted Instruction (CAI) techniques is necessary to determine which methods are readily adaptable for use with portable microcomputers. Many of the design issues normally associated with the development of CAI packages are exasperated by the current physical limitations of portable microcomputers. This thesis will discuss general CAI design issues and related compromises and trade-offs needed to successfully implement some of these designs in a portable microcomputer-based system. A systematic methodology for designing simple paired-associate learning (Drill and Practice) software is presented, which incorporates many of the lessons learned in the development of the included statistics software package.

TABLE OF CONTENTS

I. INTRODUCTION 10

II. BACKGROUND 14

  A. DEFINITION 14

  B. FIVE BASIC CATEGORIES OF CAI 17

    1. Drill and Practice 17

    2. Tutorial 19

    3. Problem Solving Tasks 20

    4. Simulation 21

    5. Instructional Games 22

  C. THE EVOLUTION OF COMPUTER ASSISTED INSTRUCTION 25

  D. OBSTACLES TO THE DEVELOPMENT OF PORTABLE MICROCOMPUTER BASED EDUCATIONAL SOFTWARE 30

    1. Technical 31

    2. Psychological 31

    3. Administrative 32

    4. Outlook 33

III. A CAI FOR AN INTRODUCTORY STATISTICS COURSE 35

  A. DESCRIPTION OF THE PROBLEM/CURRENT SITUATION 35

  B. COMPUTERS AND THE INSTRUCTION OF STATISTICS 37

    1. Presentation Mode 37

    2. Direct Mode 38

  C. STATISTICAL PACKAGES 39

D. EFFECTIVENESS OF COMMON STRATEGIES OF COMPUTER USAGE IN STATISTICS 41

IV. CAI DESIGN ISSUES 43

A. LANGUAGE SELECTION 44

1. Specialized CAI Languages 44

2. General Purpose Languages 45

B. LEVEL OF USER GUIDANCE 46

1. Menu Driven Systems 47

2. User Controlled System 48

C. LEVEL OF FEEDBACK 48

D. LEVEL OF HELP 51

E. SCREEN DESIGN 53

1. Basic Screen Design 54

2. Enhanced Screen Design 59

F. METHOD OF USER INTERFACE 60

1. Keyboard 61

2. Mouse 62

3. Light Pen 63

4. Infrared Touch Screen 63

5. Interactive Video Disk 64

6. Sound/Voice 64

V. LIFE CYCLE DEVELOPMENT METHOD FOR A DRILL AND PRACTICE TYPE CAI 68

A. FEASIBILITY ANALYSIS PHASE 71

1. Subject/Topic Validation 71

2. Development of Overall Goals and Specific Objectives 72

3. Determine Student User's Characteristics 73

4. Determine System Level of Implementation 74

B. INITIAL DESIGN PHASE 75

1. Definition of Logical Categories 76

2. Subdivision of Categories Into Smaller Groups 77

3. Determine Primary Design Features 78

4. Hardware and Language Selection 80

C. DETAILED DESIGN PHASE 81

1. Detailed Task Analysis 82

2. Module Composition and Inter-modular Flow 84

3. Initial Draft of User's Manual 86

D. PROGRAM DEVELOPMENT PHASE 86

1. Development of Algorithmic Solutions 87

2. Validating and Processing User Input 88

3. Screen and Menu Display Design 89

4. Revise User's Manual 90

E. CODE AND TEST PHASE 91

1. Individual Module Code Generation 92

2. Module Testing 93

3. Program Module Integration and Testing 94

F. IMPLEMENTATION AND COURSE INTEGRATION PHASE 95

1. Method of Student Access 96

2. Validate User's Manual 97

3. Revision of Program Package 97

VI. DESCRIPTION OF THE DEVELOPED CAI PACKAGE 98

A. OBJECTIVES 99

B. SCOPE 99

C. DESIGN METHODOLOGY AND CONSIDERATIONS 100

D. DESIGN ISSUES ENCOUNTERED 103

E. CONSIDERATIONS IN USING THE DESIGNED PACKAGE 104

F. LESSONS LEARNED 106

VII. FUTURE OF PORTABLE MICROCOMPUTER BASED CAI 109

APPENDIX A: User's Guide 112

APPENDIX B: Program Listings 214

APPENDIX C: Description of RS Model 100 282

LIST OF REFERENCES 283

BIBLIOGRAPHY 288

INITIAL DISTRIBUTION LIST 292

LIST OF FIGURES

1. Terminology Proliferation 15

2. Advantages and Disadvantages of Various CAI Strategies 23-25

3. Advantages and Disadvantages of Various User Interface Methods 66-67

4. Life Cycle Development Model 69-70

5. Distribution of Development Time by Phase 102

I. INTRODUCTION

The decreasing cost and increasing capabilities of the "truly" portable microcomputers makes them especially well suited for student use. Due to the misuse of the term "portable" by various computer manufacturers, a precise definition is in order. In this context, "portable microcomputer" refers to that class of computers which weigh less than twenty pounds, can operate for extended periods on self-contained batteries, and contain all the necessary peripheral devices for interface with the user in one compact package. This class of computer is often equipped with an internal modem and may or may not incorporate a self-contained secondary storage device. Most of these computers have a relatively small (but rapidly expanding) random access memory (RAM) capability on the order of 8-256k. Almost without exception, these computers have the ability to retain data in the RAM when not in operation.

These technological advances have made feasible the equipping of each student with his or her "own" portable computer to use for the duration of selected courses. With appropriate software, these computers can become a valuable learning tool. The likely side benefits to be derived from such individual issues of portable microcomputers include [Ref. 1]:

1. Elimination of the contention problem caused by the overcrowding of the public terminals.
2. Increased computer literacy.
3. A reduction in the "intimidation factor" perceived by many users of larger mainframe systems.
4. Complete flexibility on the part of the user to explore alternate computer applications.

The key to successful implementation of these devices in education is the development of software which has been appropriately designed to the unique characteristics of the course and microcomputer in question, and the abilities of the learner population [Ref. 2]. The field of statistics has traditionally been on the leading edge of applying computer technology to assist in the large amounts of tedious calculations involved in statistical analysis. As such, a wide variety of statistical educational software is available for use on mainframes, minicomputers and --to a lesser extent-- microcomputers. In addition, programmable calculators (e.g., the TI-59) have frequently been used to assist both students and statisticians in completing basic mathematical calculations. The development of similar packages for portable microcomputers has lagged far behind. Current limitations of portable microcomputers, particularly in the area of memory capacity, have discouraged the implementation of full scale portable microcomputer-based

CAI packages. With proper design and appropriate trade-offs, however, effective CAI packages for portable microcomputers are possible.

The portable microcomputer is too valuable as a tool for educational use to cast aside for want of creative software designs. The possibilities encompassed by individual use of portable microcomputers far exceed those of the personal calculator which has already revolutionized many aspects of basic statistical education. The software developed as part of this thesis is only a meager example of the possibilities made available by the portable microcomputer. It is hoped that this treatment will serve as the impetus for others to develop and implement much more extensive portable microcomputer-based software.

A definition of CAI and a description of the various types of strategies commonly used in the field is contained in Chapter II. A brief history of the development of CAI is also included to provide the reader with pertinent background information. The chapter closes with a short discussion of the obstacles which have hindered the full implementation of CAI techniques in general, and how they relate to portable microcomputer-based CAI software in particular.

Chapter III describes the problems which led to the development of the included software package. Various methods of computer usage in statistics education are discussed, and a review of some of the more common statistical packages is included. The last portion of this chapter is devoted to briefly assessing the effectiveness of the two most popular strategies of computer usage in statistics.

Chapter IV discusses some of the various design issues which are faced by the potential CAI designer. A wide range of issues from language selection to screen design are outlined in order to give the reader an appreciation of the complexity of CAI system design. The following chapter (V) provides an overview of a model for the development of a drill and practice type CAI. The presented model was revised and tempered during the development of the included CAI package. A description of this CAI package is provided in Chapter VI.

Chapter VII contains a discussion on the projected future of portable microcomputers in education. In addition, the major points of this thesis are reviewed and concluding comments are provided.

II. BACKGROUND

A. DEFINITION

A precise definition of Computer Assisted Instruction (CAI) has been evolving since activity in the area of computer aided instruction began in earnest in the 1960s. In early literature on the subject, the term CAI was commonly used to refer to all aspects of computers in education. As interest grew in this new and exciting field, new terms were indiscriminately introduced by various authors. Depending on the author and the context, many of these terms could be used interchangeably with CAI. Additional terms were introduced to distinguish certain applications of computers in education from the general category of CAI. The wide and rapid proliferation of terms invariably led to confusion, and different authors attached different meanings to the same terms. A sample of these terms is contained in Figure 1 [Ref. 3].

Over the last 25 years, most authors in this field have narrowed the definition of CAI and consider it a distinct subject within a broader category termed Computer Based Education (CBE). While the actual boundaries of the definition of CAI are still somewhat blurred, a general consensus of what constitutes a CAI has evolved. A

Automated Teaching

Computer-Administered Instruction (CAI)

Computer-Aided Instruction (CAI)

Computer-Aided Learning (CAL)

Computer-Aided Teaching

Computer-Assisted Education

Computer-Assisted Guidance (CAG)

Computer-Assisted Learning (CAL)

Computer-Based Education (CBE)

Computer-Based Instruction (CBI)

Computer-Based Learning (CBL)

Computer-Controlled Teaching

Computer-Directed Training

Computer-Individualized Instruction (CII)

Computer-Managed Instruction (CMI)

Computer-Managed Learning (CML)

Computer-Simulated Instruction

Computerized Instruction

Figure 1. Terminology Proliferation [Ref. 4]

composite definition of CAI representative of this consensus is:

> "CAI is a term used to describe the process of interactive teaching with the aid of a computer. The process directly involves the computer in the storage and presentation of instructional materials to the student. Both written material and graphics are presented to the student in a logical manner." [Ref. 5.]

This definition, while broad, serves to distinguish CAI from another growing field of computers in education termed Computer Managed Instruction (CMI). In CMI, the computer is not normally used as a direct interface with the student to effect learning. Instead, the computer is used as a management tool to assist the instructor. The materials presented to the student in the CMI model are not of a direct educational nature and include such things as tests and management of the flow of a student's education. The computer is used as a gate to control student progress and to provide extensive managerial data to the instructor to enable more individual tailoring of instruction.

Despite the fact that a few current authors continue to refer to both CAI and CMI as a single entity [Ref. 6], there are two important distinctions between CAI and CMI:

1. CAI is usually performed in an on-line real time or near real time basis in order to provide immediate feedback of an instructional nature to the student. CMI, on the other hand, is typically not a real time process and focuses

on the batch processing of data for later presentation to the instructor. [Ref. 7]

2. CAI is more resource intensive both in hardware and software [Ref. 8]. In order to implement a successful CAI system students must have ready access to computer terminals, necessitating a significant investment in hardware. In addition, the software (commonly referred to as courseware) required to effect a CAI is complex and long lead times and high development costs are the rule. In contrast, CMI can be effective with only a few terminals for joint instructor and student use. Software development for CMI tends to be relatively straightforward and relatively inexpensive.

B. FIVE BASIC CATEGORIES OF CAI

CAI can be further subdivided into five "teaching strategies": (1) Drill and Practice; (2) Tutorial; (3) Problem Solving Tasks; (4) Simulation; and (5) Instructional Games. [Ref. 9]

1. Drill and Practice

The drill and practice strategy is by far the easiest to develop and implement. The key to the effectiveness of this method is that the user is assumed to have already learned the basis of the material to be presented through traditional classroom lecture techniques. The goal of the computer is to cement the retention of the material by repeated practice and exercise. While new concepts can be learned using the drill and practice strategy, it is not an efficient vehicle for this type of education. This strategy does, however, provide the student with the capability to work through a large array of problems and to develop an intuitive understanding of the procedures used. [Ref. 10]

Drill and practice programs tend to be organized linearly with a brief introduction followed by the presentation of a series of problems or solution steps [Ref. 11]. A series of preselected problems may be contained within the software, or problems can be provided from outside material by the student through appropriate interfacing with the program. The latter method is more flexible and provides for more student involvement. The embedding of the actual problems in the software significantly restricts the scope of the program and requires extensive maintenance to change the problem sets. Drill and practice type software is particularly well suited to courses where a repetitive practice is required to learn a skill or a concept. The linear sequencing of solution steps for many mathematics and statistical problems can easily be accommodated using the drill and practice strategy.

2. Tutorial

The tutorial strategy encompasses a great deal more than the relatively simple drill and practice strategy. Whereas drill and practice programs serve as an adjunct to the instructor, tutorial programs strive to "replace" the instructor as much as possible. Logical dialogue between the student and the program about specific problems is the hallmark of this strategy. These programs are typically capable of detecting not just simple calculations and data input errors, but also errors in reasoning by the student. Extensive feedback is provided to the student to assist in correcting these errors. [Ref. 12]

These programs are exceedingly complex to design and develop, requiring an extensive assortment of programming techniques. Up to and beyond 200 manhours are often necessary to achieve only one hour of student instruction [Ref. 13]. In addition to the initial developmental expenses, maintenance costs for this strategy are correspondingly high. The complexity of the programming task often leads to the selection of a "custom" programming language tailored to educational software to simplify the task. Unfortunately, the use of these "custom" languages severely restricts the portability of the finished software and, therefore, further inflates the cost of such a system.

The tutorial strategy is most often viewed by the layman as the epitomy of what a CAI should be. The vast majority of the early large scale endeavors in the field of CAI were of this type. As computer technology has advanced, improvements have been incorporated into the tutorial strategy and it remains the "cadillac" of CAI methods. This technique can be tailored to fit most any learning situation but it excels in imparting complicated conceptual skills. [Ref. 14]

3. Problem Solving Tasks

This strategy encompasses programs which are written to solve certain select problems in a specific area. This is the least well defined of the five strategies presented in that a wide variance of sophistication can exist between different programs within this strategy. This strategy places no boundaries on the various applications or level of difficulty of problems. In general, these programs tend to be more "solution oriented" and less concerned with developing computational or conceptual skills of the student. Most applications within this strategy parallel the use of a simple calculator; however, in this case, the programs have the ability to solve problems of far greater complexity. This strategy is well suited to quantitative courses which are primarily interested in the final solution and not the methods or techniques to derive the solutions. [Ref. 15]

4. Simulation

This strategy is generally used for the generation and manipulation of data or the repetitive cycling of a model when environmental factors preclude normal methods. These environmental factors include limitations of time, money or equipment; and various safety considerations.

This strategy is based on the mathematical manipulation of complex models which places the student in a controlled "real life" situation. Most programs of this type allow the user to input a series of parameters and process these, in a compressed fashion, through the model. The resultant information is displayed for user perusal and may include pertinent comments to key the student to significant data. In general, no learning "path" is predefined. Instead, the student is allowed to learn through actual manipulation of these processes. These types of programs are frequently used in statistical analysis and can be very effective learning tools. Concepts learned via this strategy are likely to be retained by the student longer. [Ref. 16]

While the costs to design and develop programs of this type are not overly prohibitive, the cost to execute them on a large scale is typically very high. A moderately sized simulation may require an inordinate amount of computer resources to execute. For these reasons, many simulations are done in advance by the instructor and presented in the traditional lecture format. [Ref. 17]

5. Instructional Games

The distinctive feature of this strategy is an attempt to use a student's competitive nature to achieve certain learning goals or skills. These programs most often pit the student against the computer in a win-lose situation in an effort to hold the student's attention and interest. Creative programming skills are required to make this strategy effective as a learning tool. This technique is ideally suited to elementary level education.

Each of the five strategies presented above has a series of advantages and disadvantages associated with them, some of which are obvious and others which were only discovered through actual applications of the strategies. A summary of these advantages and disadvantages is provided in Figure 2. The most significant problems of complexity and cost have already been mentioned while other problems will be brought out in the historical perspective of the evolution of CAI that follows.

	Advantages	Disadvantages
Drill & Practice	*Easy to develop and implement. *Allows a student to work through a large number of problems to gain an intuitive understanding of the solution procedures. *Performs well with problems which have a well defined linear sequencing of solution steps. *Capable of being implemented on relatively inexpensive microcomputers.	*User required to already have learned the basis of the material through other methods. *Inefficient for teaching new concepts. *Requires the student to "learn" the subject material through repetitious use of the program. *Normally not capable of detecting errors in reasoning.
Tutorial	*Can be designed to reduce instructor workload, or even to replace the instructor. *Capable of detecting errors in student reasoning. *Can be tailored to almost any learning situation. *Performs well in teaching new or complicated conceptual skills.	*Complex to design and implement. *High design cost. *High maintenance cost. *Requires extensive computer hardware support.

Figure 2. Advantages and Disadvantages of Various CAI Strategies

	Advantages	Disadvantages
Problem Solving Tasks	*Relatively cheap to design. *Uses a minimum of computer resources. *Can be used effectively with devices as simple as programmable calculators. *Provides a rapid method of obtaining the solution to a problem.	*Solution orientation does not allow for the development of student computational or conceptual skills. *Relatively inflexible. *Often used by students as a "crutch" not a learning tool.
Simulation	*Can be used to create an artificial environment to obtain data about a situation or process which otherwise would be impractical to study due to limitations of time, money or safety hazards. *Student learning is accomplished through actual manipulation of the parameters of various processes. *Typically results in longer retention of learned concepts.	*Requires extensive computer resources to execute. *Not practical for simultaneous use by a large number of students. *Can overwhelm the student with large volumes of data. *Can require the student to learn various computer techniques unrelated to the subject matter.

Figure 2. Continued

	Advantages	Disadvantages
Instructional Games	*Harnesses a student's competitive nature to accomplish learning objectives. *Can add a degree of excitement to learning otherwise dreary topics.	*Requires creative programming skills. *Difficult to develop programs which will continue to challenge and hold a student's interest. *Students set high expectations based on their experience with video games.

Figure 2. Continued

C. THE EVOLUTION OF COMPUTER ASSISTED INSTRUCTION

Before launching into a discussion of the evolution of yet another technology based teaching aid, it is appropriate to put the use of the computer in education in perspective by reviewing some predictions made about previous technological "miracles".

The widespread introduction of movies in the early 1940s led many predictors to make some astounding predictions. These predictions are best summed up by a 1942 quote from Charles F. Hoban, Jr., who was then a member of the American Council on Education:

> "The motion picture is the most revolutionary instrument introduced into education since the printing press." [Ref. 18]

The test of time has indeed proved the worth of movies to education but not anywhere near the level Mr. Hoban was suggesting. The extensive use of educational movies on a daily basis in public schools did not come to pass. Similar predictions were made about television in the late 1950s, and about programmed learning in the early 1960s.

> "Television offers the greatest opportunity for the advancement of education since the introduction of printing by moveable type." [Ref. 19]

> "Programmed learning is the first major technological innovation in education since the invention of printing." [Ref. 20]

These quotes are presented not to ridicule their authors, but rather to provide an added dimension in which to consider similar claims which were made about CAI in the 1960s.

Computer assisted instruction is more than 25 years old. Although conceived in the late 1950s, CAI did not really begin to expand until the early 1960s. The earliest CAI applications occurred quite naturally in the computer industry as a method of employee training [Ref. 21]. A virtual explosion in the amount of literature pertaining to various CAI techniques occurred during this time period as the use of the computer as a learning tool captured the imaginations of authors and educators. A series of related collateral events helped to spur the creation of CAI software. The most important of these was the introduction by IBM in 1960 of the first "author" (programming) language, COURSEWRITER, which was tailored to support CAI applications. COURSEWRITER and a series of other specialized languages enabled people with a lower level of programming skills to develop moderately sophisticated CAI programs. Previously, CAI programs had been coded in complicated languages, such as FORTRAN, which were very difficult for a "non-programmer" to fathom. [Ref. 22]

The opening of CAI design and development to nonprofessional programmers had a profound effect on the expansion of CAI. A slew of CAI software soon became available on everything from ecology and driver education to nuclear war [Ref. 23]. Most of these early projects were tutorial in strategy and made no clear distinction between CAI and CMI. Some of the more significant products of this period included systems such as PLATO and TICCIT. The PLATO system was one of the first large scale CAI projects when it was started in 1960 at the University of Illinois, Urbana-Champain. A great deal of funding was allotted to the PLATO project by a variety of federal, state and private activities. Contributors included the Advanced Research Projects Administration, the National Science Foundation, and the Control Data Corporation. The primary goals of the PLATO project were:

> ". . . to produce a cost-effective computer-based educational system with the power and flexibility necessary for provision of high quality interaction." [Ref. 24]

The technological achievement of PLATO in CAI design and implementation was considered a benchmark in the developmental history of computers in education.

TICCIT (Time-shared, Interacting, Computer Controlled Information Television) was developed in 1971 by the MITRE Corporation in conjunction with the University of Texas and Brigham Young University [Ref. 25; Ref. 26]. It was conceived as a smaller and less expensive alternative to larger CAI systems. TICCIT made extensive use of the technology of the period incorporating minicomputers and a host of CRT terminals in a time-shared mode. Military applications of CAI technology have also made significant contributions to the field. The U.S. Navy, for example, has made extensive use of an adaptation of the PLATO IV system and a version of the TICCIT system to assist in aircrew training [Ref. 27].

By the mid-1970s, interest in CAI began to wane. The overriding factor for this fairly rapid decline was that despite a few convincing successes, CAI in general had failed to live up to the lofty expectations that had been thrust upon the field by early writers and educators. Cost is the most frequently cited reason for this "failure". Although hardware costs continued to decline during this period, the cost of suitable software increased dramatically. Software developers were concentrating on the design and production of costly tutorial type CAI systems.

In addition to being costly, the tutorial systems had a limited market, exasperated by machine dependencies. Consequently, no single manufacturer was successful in developing "a stable profit-making product line" [Ref. 28]. Other factors which contributed to the decline in popularity of CAI systems included the attitude and training of teachers. Teachers were not properly trained in the use of CAI systems and were not fully aware of the capabilities of such systems. This unfamiliarity with CAI systems often resulted in the development of a hostile attitude toward the use of computers in the schools [Ref. 29].

While CAI applications were being drastically scaled down, the use of the computer to support Computer Management Instruction (CMI) gained wider acceptance in the educational community. CMI has two distinct advantages over CAI: (1) it is less costly, and (2) it is less threatening to educators. The reduced costs of a CMI are the result of a reduction in hardware costs and the relatively simple software requirements. The orientation of CMI toward providing administrative help to educators, vice becoming directly involved in the teaching process, engendered a more positive attitude for teacher acceptance of the computer.

While CMI techniques continued to expand, interest in CAI systems underwent a resurgence with the widespread proliferation of personal microcomputers in the late 1970s and early 1980s [Ref. 30]. The popularity of microcomputers has opened new markets for specialized microcomputer compatible educational software. The reduced memory capacity available with microcomputers led to the more widespread use of alternative strategies to the traditional tutorial mode. Drill and practice type CAI seem ideally suited to these new machines and this strategy has rapidly become dominant in terms of the numbers of applications [Ref. 31].

Subsequent development of CAI systems through the present time has continued to reflect an emphasis on designing new systems or adapting older systems for use on networks of microcomputers. The development of larger, more capable CAI systems for use on mainframe and minicomputers continues and is incorporating many of the features previously unique to CMI. The designers of CAI systems have turned to professional educational psychologists and are studying various learning theories in an effort to enhance the effectiveness of their products.

D. OBSTACLES TO THE DEVELOPMENT OF PORTABLE MICROCOMPUTER-BASED EDUCATIONAL SOFTWARE

The development of educational software for portable microcomputers has continued to lag far behind that of its microcomputer big brothers. This section discusses technical, psychological and administrative obstacles which have hampered widespread implementation of portable microcomputer-based CAI software.

1. Technical

The physical limitations of the current generation of portable microcomputers places certain constraints on the selection of a CAI strategy. The lack of an abundant amount of random access memory (RAM) is the most frequently cited limitation, but other factors such as the use of less capable display screens (no color, limited graphics, limited size) and a lack of viable secondary storage also contribute to the problem. A related issue is the inability of most portable microcomputers to support a wide array of languages. This limitation severely limits the transferability of software to portable microcomputers which was designed initially for use on larger systems. The portability of educational software in general is viewed by some authors to be the largest obstacle to be overcome by CAI, and the introduction of portable microcomputers merely exasperates this situation. [Ref. 32]

2. Psychological

The lack of a complete acceptance of general CAI techniques by educators continues to be a hinderance to the development of CAI systems. The concentration on drill and practice type CAI applications for use with portable microcomputers can potentially eliminate much of this educator resistance. Other more subtle problems continue to persist, however. Educators are reluctant to redefine the overall objectives or restructure a particular course in order to facilitate the use of portable microcomputers. Instead, use of portable microcomputers is tacked onto the existing course curriculum and no real thought is given to a deeper integration of the course and the computer. In many ways, the portable microcomputer is thought of, and thus used, as if it were merely another calculator. Until a more thorough integration of portable microcomputers with course material is carried out, the effectiveness of this method will be limited. [Ref. 33]

The introduction of any computer-based method of instruction into a classroom necessitates a re-evaluation of the role of the instructor and the responsibilities of the student. The introduction of a large number of portable microcomputers for individual student use can contribute to a feeling on the part of the instructor of a lack of control of the educational situation.

3. Administrative

Certain basic administrative problems manifest themselves with the widespread introduction of a portable microcomputer-based CAI. To be effective, student to computer ratios should be as low as possible, preferably one to one. Thus, in implementing such a system, a method of distributing and accounting for a large number of portable microcomputers is required. In addition, while the new generation computers are designed to withstand a fair amount of rough handling, a system to provide for the maintenance and upkeep of these devices must be developed.

The proceeding comments serve to point out the necessity of developing detailed plans in order to successfully integrate the use of portable microcomputers into a curriculum. Bureaucratic administrations tend to be reluctant to develop such plans, preferring to opt for the path of least resistance, and maintain the existing banks of non-portable microcomputers which are better established and are under more direct supervisory control.

4. Outlook

Continuing technological advances may effectively eliminate many of the physical obstacles cited above. However, it is more difficult to assess the importance of the psychological and administrative problems. Nonetheless, as the use of portable microcomputers becomes more widespread, many of these problems will fade. Perhaps the largest "cloud on the horizon" is rooted in the premise that individualized instruction is the best learning technique. The very heart of CAI is based on this premise, which has recently come into question. In situations where a predefined set of skills must be learned in a relatively short period of time, individually paced instruction is of less value. [Ref. 34]

In order for most CAI strategies to remain fully effective in these situations, methods must be developed to transform inherently individualized CAI programs into a system capable of treating a large number of students as a single block. Due to its simplicity in design and purpose, the drill and practice type of CAI can more easily accommodate these scenerios.

III. A CAI FOR AN INTRODUCTORY LEVEL STATISTICS COURSE

A. DESCRIPTION OF THE PROBLEM/CURRENT SITUATION

Statistics has traditionally been on the leading edge of the application of computer technology due to the excessive amount of routine (and often repetitive) calculations necessary to complete many statistical procedures. The professional statistician has learned to rely on the power and flexibility of the computer and would be lost without it. In the use of advanced technology, the instruction of introductory statistics has centered on the use of programmable "smart" calculators, such as the Texas Instruments TI-59, or on the integration of mainframe-based statistical packages, like SPSS. Both of these methods have associated drawbacks which could be eliminated by the introduction of portable microcomputers.

"Smart" calculators, while invaluable tools to the educated user, are much less useful to the uninitiated student as a learning aid. The procedures for interfacing with a programmable calculator usually differ significantly from normal thought processes which would be involved in solving a problem by hand. The end result is often a less than optimum understanding of the problem solving process and a mindless memorization of keystrokes which fails to impart an intuitive understanding of the problem. In many situations, the calculator becomes a crutch, thoroughly negating the educational benefits for which it was originally introduced.

The integration of mainframe-based large statistical packages into the curriculum of introductory statistics courses has indeed eliminated many of the problems attributed to the use of the programmable calculator. These statistical packages have, however, introduced new problems. The singular use of any one of these packages requires a significant investment in computer resources. Multiple simultaneous users of one of these packages can put an unacceptable load on these expensive computer resources. The user is required to learn the Job Control Language (JCL) and processing characteristics of large, complex host machines in addition to the normal course requirements. These packages are, in general, far more sophisticated than is required for instruction of an elementary statistics course. A student can be overwhelmed by the amount of information generated by a single run of one of these packages. The inclusion, by these packages, of data which has no meaning for the introductory level student can cloud the basic issue and can easily intimidate the less capable learner. Finally, the accessability and availability of computer terminals to interface with these packages can become a major problem.

With the use of appropriate software, the portable microcomputers currently available possess the capabilitiess to overcome many of the problems which have been associated with the above systems. Competitively priced portable microcomputers make it feasible to provide each student with his or her own unit. This expands the accessability and availability of portable microcomputers to the same level as programmable calculators. The stand alone operation of these units eliminates the expensive consumption of mainframe computer resources. Nevertheless, most portable microcomputers have the ability to interface with the mainframe over standard telephone lines if necessary. A properly designed portable microcomputer-based CAI can provide the introductory statistics student with tailored educational support, and reinforce the course learning objectives.

B. COMPUTERS AND THE INSTRUCTION OF STATISTICS

Two distinct modes of computer usage are commonly used in the instruction of basic statistics [Ref. 35].

1. Presentation Mode

This technique is often used when there are inadequate computer resources on hand to support a large number of simultaneous users. It consists of one person, usually the instructor, using the available computer resources to demonstrate a particular statistical procedure or problem in either an online or offline basis. If used in an online basis, the instructor will typically execute the demonstration in the presence of the class in a room which has been equipped with additional television monitors for ease of student viewing. This method allows the student to actually see the computer in operation. When used in an offline basis, the instructor will obtain a hardcopy record of a session for later presentation to the class. This latter method has the advantage of enabling the instructor to edit the results so that it conforms exactly to the subject topic.

2. Direct Mode

This technique is widely held in literature as superior to the presentation mode; however, its use requires a significant amount of computer resources. This method requires each student to individually interface with the selected statistics package. This package can be mainframe-based or microcomputer-based. If microcomputer-based, the interface is almost universally accomplished on a real time basis. If a mainframe package is chosen, then the student's interface may be on a real time interactive basis through the use of remote terminals, or on a non-real time basis (offline or batch processing). The advantages of this mode are that it enables each student to individually manipulate data and, in most cases, maintains a higher degree of student interest.

C. STATISTICAL PACKAGES

A wide variety of statistical packages have been developed over the last 25 years. Most of these were developed for use by the professional statistician and are of little use to the uninitiated student. Programs that have been developed or converted for student use fall into two main categories--mainframe/minicomputer-based packages and the microcomputer-based package. Individual packages from the first category have found ready acceptance from a substantial number of institutions which are equipped with adequate computer resources to use these large packages effectively. No single microcomputer-based package has received comparable broad spectrum acceptance; however, they outnumber the mainframe/minicomputer-based packages [Ref. 36]. Contributing to the expanding numbers of microbased statistical packages is the propensity for many, if not most, institutions to develop their own programs tailored to local needs. The introduction of relatively inexpensive microcomputers has also contributed to this proliferation of tailored software. On the other hand, the complexity and expense of developing large packages of interactive statistical software for mainframe systems has tended to favor a few general purpose statistical packages.

Representative of the mainframe/minicomputer-based statistical packages are the Statistical Package for the

Social Sciences (SPSS) and Minitab. SPSS is designed for use on large mainframe computers and consists of a

> ". . . comprehensive software collection which form a library of analytical techniques and statistics most often used in business and data processing. Its chief advantage includes a uniform organization of data for all types of analyses and simple command statements that can be easily and rapidly changed to meet most common designs". [Ref. 37]

Minitab is designed for use on a medium size computer and is a student adaptation of an earlier system called OMNITAB. Minitab affords the student the capability to converse with the computer in commands similar to English. One advantage of Minitab is that it is written in FORTRAN II and is easily adaptable to a wide range of medium to large screen computing systems. [Ref. 38]

Since no single microcomputer-based statistical package has developed anywhere near the acceptance of the larger statistical packages, it is more appropriate to discuss current overall microcomputer implementation methods. One method commonly used in the teaching of statistics with the aid of a microcomputer is to have the students self-generate solution algorithms using a high level programming language. A second approach is to provide the student with "canned programs" to use on the microcomputer at his leisure. These programs, whether student authored or canned, can cover the whole range of CAI strategies from drill and practice to simulations. In most situations, however, these programs tend to be of the drill and practice or problem solving type due to the larger resource requirements of the other CAI strategies [Ref. 39; Ref. 40]. (One exception to this is that microcomputer compatible versions of some of the larger mainframe-based packages like SPSS and Minitab have recently become available. These programs are a much abbreviated form of the original package, but retain the capability to conduct most statistical calculations.)

While the first method of student generated solution algorithms has the advantage of providing a more thorough understanding of the mechanics of obtaining a solution, it requires the students to have a working knowledge of a programming language. Since most introductory statistics courses do not require previous programming experience, this method is impractical for most applications.

D. EFFECTIVENESS OF COMMON STRATEGIES OF COMPUTER USAGE IN STATISTICS

The CAI strategies of simulation and drill and practice have consistently been singled out in the literature as the most effective methods of assisting in statistical education [Ref. 41; Ref. 42]. Simulations are particularly effective because they enable the student to manipulate a large number of variables to produce varying results. By a judicious comparison of the input variables and the generated output from a series of runs, a student can gain a more thorough understanding of the relationship of statistical theory to actual real world situations. These simulations also serve as a practical, economical and safe way to expose students to a variety of situations which can enhance confidence in statistical methods. The major drawback of such simulations is that in most cases, they require the support of a large computer system.

The drill and practice strategy has the advantage of being capable of implementation on almost any computer device. This strategy is particularly well suited for use on portable microcomputers. The objectives of the drill and practice strategy are much more narrowly defined than most other CAI strategies. This strategy relies on the repetitive solving of a variety of different problems from one category to cement the procedures in the student's mind. Classroom time is typically consumed in providing theoretical background and in stepping through one or two examples. The drill and practice type CAI provides the student the capability to work through a large number of similar problems in a reasonable period of time. By freeing the student from most of the time consuming mathematical computations, this method allows more time for concentrating on developing an in-depth understanding of the solution process. When implemented on individually issued portable microcomputers, the student is relieved of the requirement to complete these problems in the artificial environment of a school's computer terminal room. [Ref. 43]

IV. CAI DESIGN ISSUES

This section addresses various design issues involved in CAI program development. This discussion is intended to be general in nature and is not constrained by any particular hardware configuration. It is important to remember, however, that the availability/selection of computer hardware is a major driving factor in CAI design. It makes no sense, for example, to develop a CAI package which relies on sophisticated color graphics if the available selected hardware cannot support color graphics. It is imperative, therefore, that the CAI designer or design team know in advance the general capabilities and limitations of the system their product is to be implemented on.

CAI design issues can be divided into six categories: (1) language selection; (2) level of user guidance; (3) level of feedback; (4) level of online help; (5) screen design; and (6) method of user interface. While each of these categories is hardware dependent to some extent, the method of user interface exhibits the greatest hardware dependency. The other categories have greater flexibility in accommodating a variety of hardware configurations. The various alternatives available within each category are briefly discussed in the following paragraphs [Ref. 44].

A. LANGUAGE SELECTION

The selection of a programming language to use in implementing a CAI is driven by two major factors: the ability and programming skills of the potential CAI developer or builder and the capabilities of the implementing system hardware. The development of CAI packages has typically followed two paths which are directly related to the above factors--the use of specialized CAI language or the use of general purpose high level languages. (Early CAI packages which were based on a complex menagerie of low level and machine languages have largely been supplanted by updated versions which follow one of the above two paths.) [Ref. 45]

1. Specialized CAI Language

The introduction of special purpose CAI languages greatly expanded the popularity of CAI systems. These languages enable people with little or no prior programming experience to design effective CAI packages. The two premier examples of such specialized languages are IBM's COURSEWRITER, and TUTOR for use on PLATO systems [Ref. 46]. The structure and syntax of these languages is straightforward and special provisions have been made to easily accommodate feedback loops and error detection/correction. The major drawback of such languages is the limited number of systems which can support their use. Most of these specialized languages have been developed for use on large to medium size computers. Even portability between similar size computers can be a major problem, and microcomputer uses are usually not possible. These languages are, for the most part, optimized for conversational uses and do not support intricate mathematical calculations as well as most general purpose languages [Ref. 47].

If the proper hardware is available and portability is not critical to the economic success of the finished product then specialized CAI languages are the path to take. This is especially true if the author's programming experience is low and if the programs do not entail excessive mathematical computation.

2. General Purpose Languages

There currently exists a wide variety of general purpose languages which are suitable for the implementation of a CAI package. Two of the most popular are BASIC and APL [Ref. 48]. The use of BASIC as a CAI language has been accelerated by the introduction of microcomputers, which as a rule have the capability to support this relatively simple nonstructured language. The use of BASIC has become so widespread (BASIC became the most frequently used CAI language in 1976 [Ref. 49]) that current versions have been provided with control syntax to support structured programming.

The proper use of common general purpose programming languages can greatly enhance the portability of the finished product. In addition, the expanded computational capabilities of languages such as APL make them particularly well suited for tasks involving intricate calculations. The major drawback of such languages is that their use requires a certain amount of training and expertise. Experience has shown that in order to use these languages as effective vehicles to implement a CAI, a detailed working knowledge of the language is required.

In many cases, the selected hardware will require the use of a general purpose language. This type of language is always a good path to follow if the author is an experienced programmer or if portability of the finished product is required.

B. LEVEL OF USER GUIDANCE

A great disparity exists between different CAI packages as to the amount of guidance provided to help navigate the user through the program. The level of guidance must be thorough enough to enable the inexperienced user to access the system, but concise enough not to slow down experienced users. Most literature on the subject centers on the use of menu driven systems where at each stage of a program, the user is provided with a restricted set of options. There is, however, a less frequently used method which grants more autonomy of control to the user. These "user controlled" programs allow the user to branch anywhere on the program at any time. Both methods embody significant issues which need to be addressed [Ref. 50; Ref. 51].

1. Menu Driven Systems

The use of various menus to guide a student through a CAI program presupposes a finite number of solution paths, each with well defined sequencing. This assumption holds true for most CAI applications, particularly in mathematics and statistics. Well designed menu displays can eliminate the need for extensive external user documentation and can compress the amount of time necessary to learn to interface with a program. Menu driven systems can be subdivided into single and multi-level menu systems. A single level system is designed with one basic set of menus which are viewed by all users. The multi-level system is comprised of a number of groups of menus, with a different group of menus designated for use by different categories of users. The advantage of the multi-level system is that it allows a more experienced user access to a "streamlined" set of menus to speed execution of the program while still providing a group of menus containing more detailed guidance to the less experienced user. Multi-level systems, however, require additional development time and computer overhead when compared to the less flexible single level systems [Ref. 52].

2. User Controlled System

User controlled systems provide more autonomy to the user in determining the flow of the execution of a program. This type of system is well suited to presenting lesson material in which sequencing is unimportant [Ref. 53]. The complexity of designing user controlled systems has limited their usefullness to date. Another problem associated with this method is that the user must have a thorough knowledge of the structure of the program in order to know all the branching options which are available. In addition, by allowing for total user control, the student may spend an excessive amount of time reviewing material which has already been covered [Ref. 54].

C. LEVEL OF FEEDBACK

The ability of a CAI system to provide instant feedback to the user is perhaps the most valuable aspect of the utilization of computers in education [Ref. 55]. In the context of CAI systems, feedback refers to program generated information which is provided to assist the user in achieving educational objectives and to supply guidance in the proper execution of the program. Feedback is routinely used to alert the student to the use of improper procedures or incorrect solutions, to request additional student input, and to provide appropriate comments to enhance a student's confidence or to reinforce certain aspects of learning.

This feature sets CAI apart from the earlier, more traditional methods of individualized instruction such as programmed instruction. With the use of a computer, the system has control of the feedback and can tailor it for display at precisely the right moment to optimize its effectiveness. Programmed instruction, on the other hand, includes no such control mechanism.

All five of the previously mentioned CAI strategies include a method to provide feedback; however, the content and the amount of feedback presented to the student varies among the different strategies and, to a lesser extent, between applications within the same strategy. Programs based on the tutorial strategy, for example, typically provide a far greater range of feedback than the simpler drill and practice type CAIs. To illustrate this point, consider the situation when a student inputs an incorrect solution to a question generated by a CAI program. Typical feedback commonly encountered by a student using a drill and practice type CAI in this situation may consist of a relatively brief statement alerting the student to the error and perhaps providing the correct solution. In addition to this infoormation, the feedback provided by a CAI program based on the tutorial strategy may point out possible alternative solution methods, and in the more sophisticated programs, a diagnosis of the student's error may be available. The problem facing CAI designers is to decide

(1) at what point in a program is feedback appropriate; and (2) what should the feedback consist of?

It has long been an assumption of educators that feedback facilitates education and (in its simplest form) the more constructive information contained in the feedback, the better [Ref. 56]. This assumption, which is exceedingly difficult to prove definitely, harbors serious implications in the design of a CAI system. As the desired level of feedback within a CAI increases, program design becomes more difficult. The additional control mechanisms required to implement a high level of user feedback can add significantly to the time and cost of development. The use of a higher level of feedback also consumes more computer overhead, particularly memory space. This additional draw on computer resources is typically insignificant for moderately sized CAI systems which are implemented on large mainframe computers. In the development of CAI software for microcomputers, however, the increase in overhead requirements can become a critical issue.

In order to minimize the amount of precious computer resources dedicated to providing feedback within a CAI, it is necessary to better define what the optimum level of feedback should be. (Notice that this statement rejects the assumption that more feedback is automatically better.) The results of various studies in the literature which attempt to answer this question are inconclusive. W. J. Roper conducted a quantitative analysis of the problem in 1977 using a statistics oriented CAI as the test vehicle [Ref. 57]. Quantitative testing of students who had been exposed to the same basic CAI package but with various levels of feedback demonstrated that "information-providing" feedback (such as providing the correct answer) was superior to feedback which merely let the student know whether his input was correct or incorrect. The study, like previous studies, failed to quantify a level of feedback beyond which is counterproductive. It is nonetheless intuitive that such a level exists.

The proper incorporation of feedback in a CAI is an exceedingly complex issue and even seemingly minor points require a significant amount of thought. However, because of the importance of feedback to the success of any CAI, the design team must make every effort to determine and to provide an appropriate level of feedback.

D. LEVEL OF HELP

HELP is the general term used to describe materials which are provided to assist the user in understanding the operation of the program or the content of the covered material. Many of the same issues which were discussed about feedback apply to the level of HELP. A major difference, however, is that HELP can be provided offline, as well as online. Offline HELP can consist of users manuals, student guides, textbooks, etc. It is up to the designers to determine a proper balance between online and offline HELP materials. Offline hardcopy materials tend to be cheaper than the development of online HELP facilities, and they have the obvious advantage of not consuming any computer overhead. Online HELP has the advantage of always being readily accessable to the student during a session.

A number of issues affect the complexity and cost of designing online HELP materials. The ability to invoke HELP from any point in a program and then resume processing from the point where HELP was invoked is desirable, but it entails the consumption of significantly more computer resources than simpler methods. Some programs, for example, will allow you to branch to the HELP display at any time but processing is resumed at any one of a number of set return points. Obviously, as the number of set return points increases, this method more closely approximates the prior method. The indiscriminant addition of these set return points beyond a certain level will, however, eliminate any savings in overhead and development costs. Perhaps the simplest method of providing online HELP is to have a limited number of predefined points where a user can invoke HELP and always resume processing at the beginning of the program.

Offline HELP can significantly enhance the effectiveness of a CAI package if it is carefully constructed.

Unfortunately, the development of operations manuals and other HELP materials has traditionally been neglected and, in many cases, has become a task which is not started until the programming phase has been completed. The use of menu driven systems has reduced the need for offline HELP in many systems but a certain minimum level of requirements for offline HELP should be maintained. As a minimum, the procedures for loading and executing the program, along with a brief description of the purpose and methods used, are required. Examples of valid input and the corresponding output can be particularly helpful to the beginning student and should form a large part of any offline HELP documentation.

E. SCREEN DESIGN

Screen design is concerned with the proper display of information to the user. The term "screen" refers to the device used to display the information, typically electronic monitors or LCD displays. It is a broad category in which decisions affect how information is formatted for display throughout the program, including such items as menus, feedback and HELP information, as well as educational materials. For the purpose of this discussion, screen design issues have been separated into two subcategories: (1) basic screen design, and (2) enhanced screen design. Basic screen design includes such topics as screen control, text appearance and format, and character appearance. Enhanced screen design encompasses the use of color and graphics in CAI systems.

1. Basic Screen Design

a. Screen Control

Screen control is concerned with the method of determining how long information should appear on the screen and how information should be replaced or refreshed with new information. Most CAI systems control the screen display through a combination of scrolling, scrolling with pause, and paging [Ref. 58].

--Scrolling: This method approximates what a user would see with an old teletype system. New information enters the display on a line-by-line basis either from the top or the bottom of the screen as old information disappears off the opposite side of the screen. (Alternately, information can be scrolled horizontally across the screen on a character-by-character basis. This method is not as frequently used as vertical scrolling, but can be an effective means of communicating special messages to the user.) Determining the rate at which this scrolling proceeds is critical; too fast and the student will not have time to absorb the material, too slow causes unnecessary delays in the program and may lose the attention of the student. While most all systems are capable of supporting scrolling, some can perform the task better than others.

The better systems provide for a smooth, continuous flow of information, while in other systems, the movement of information appears to be jerky and, consequently, is harder to read. As a general rule, scrolling should stop when there is no additional information to display, leaving the last screen full of data available for perusal.

Scrolling information is particularly appropriate for the display of long columns of data which cannot be accommodated by a single screen. Scrolling is also appropriate when the system requires students to respond to a series of related questions in stepwise fashion. In this manner, the student is afforded the opportunity to view the immediately preceding questions and his or her responses.

--Scrolling With Pause: This method embodies all the capabilities of normal scrolling with the additional feature of allowing the student to momentarily stop the scrolling action at will. This addition increases the flexibility of any CAI and allows for a much wider range of reading comprehensive ability. A slow reader is now afforded the opportunity to control the overall speed at which information is presented. This feature is especially useful when mathematical formulas or complex data are displayed. Many students require more time to absorb this type of material and the ability to suspend the scrolling action is one user controlled method of allowing for this extra time. This method also has a psychological advantage over regular scrolling in that the pause feature gives the student a feeling of control over the program.

--Page Control: With page control, new information enters the display in blocks vice line-by-line as in vertical scrolling. These blocks of information (often termed a page) are usually designed to fill one complete screen. In most cases, the "refreshing" of the screen is controlled by the student and in some programs, the student can advance this page forward as well as backward. This feature is extremely useful in reading textual material such as in the HELP mode of most CAIs. With paging, a designer can individually format each screen full of information to optimize the visual effect of the material. Paging eliminates some problems commonly associated with scrolling, such as scroll rate, and is usually easier to read. Paging is the preferred method for most purposes in CAI [Ref. 59].

b. Character Appearance

Character appearance addresses the issue of exactly what type of characters should be used to display information. To some extent the type of characters available will be restricted by the supporting hardware, but most newer hardware configurations can support a wide range of character fonts. This discussion will center on the methods most commonly used in existing CAI systems--uppercase only; mixed case; and special fonts.

--Upper Case Only: In some of the older computer systems and in a few of the less expensive personal microcomputers, upper case letters are the only option available. (If the CAI being designed is to be implemented solely on such a system, the designer should ignore this section.) Most computer systems today support both upper and lower case, and the judious use of all caps in certain select areas can enhance a screen display. In addition to highlighting titles, uppercase characters are effective in emphasizing certain key points in a lesson or in providing a ready means to visually separate user input from system feedback. The types of items which are going to be displayed in uppercase characters should be decided upon early in the design of a CAI so that their use is uniform through all applications. The random use of caps is pointless and confuses the student.

--Mixed Case: Mixed case is the converse of upper case only. Mixed case characters make text easier to read and, as a general rule, should be used for this reason. A particularly effective method to highlight important material is to embed mixed case characters within surrounding material which is in all caps.

--Special Fonts: Special fonts in this context include such character formats as italics, bold, Roman, and underline. These character fonts are all used to draw attention to a certain point or topic. To be effective, they should be used sparingly. Because of the hardware restrictions of many systems, it would be prudent to determine the capabilities of the proposed implementing system before special fonts are included as part of the design of a CAI.

c. Text Appearance and Focus

Text appearance is the formatting of the display of sentences and paragraphs. Issues within this category include line spacing, and the proper logical focusing of content and quantity.

--Line Spacing: Textual data is generally easier to read in a double spaced format. Single spacing can be an effective method to separate certain select blocks of information. When deciding on a choice of line spacing, the size of the screen should be considered. If the screen can only display a relatively small number of lines at one time, double spacing may be out of the question. (Most CAIs are designed for the larger computers and microcomputers which have a full size screen capability.)

--Logical Focus of Content and Quantity: Logical focus of content and quantity is a cryptic way of pointing out that the basic rules of good writing should be followed in the design of a CAI. Thoughts or ideas should not be split between displays. The quantity of information contained on any one display should be sufficient to cover the lesson topic at hand but not overwhelm the student. The screen should not appear cluttered or overcrowded [Ref. 60].

2. Enhanced Screen Design

a. Color

The proper use of color makes any program more attractive and can be a particularly effective CAI technique. The basic issue with the use of color is when and where it can be used most effectively. Color is most effective at separating portions of the screen display to designate certain areas for specific uses; feedback only, for example. Special modules within a program can be assigned different background colors to distinguish them from normal program operations; the HELP display, for example. The use of color to display text is an effective method of highlighting certain key points. Color should not be used to such an extent that it draws attention away from the topic subject. In addition, it should be remembered that certain colors can increase eye strain if used to display textual data. Remember, color is nice, but most topics which are readily adaptable to CAI techniques can be effectively programmed without the use of color. This is not to say that color should not be used, but merely to point out that the introduction of color should serve a well defined purpose in the program [Ref. 61].

b. Graphics

The increasing capabilities of almost all types of computer systems to generate high quality graphics makes their incorporation into even the simplest CAI system more likely. To be used effectively, they must be carefully tailored to individual topics within a CAI system. Since the primary focus of this thesis is on statistical CAI systems, it is interesting to note a disparity in the literature regarding the use of graphics in statistical applications. While most authors enthusiastically endorse the use of graphics for illustrating statistical concepts, the only quantitative study located on the subject for this thesis found no significant difference between CAI systems with graphics as opposed to those without graphics [Ref. 62]. What this suggests is that CAI designers should use graphics sparingly and only when an illustration is likely to add additional information not readily discernible from text.

F. METHOD OF USER INTERFACE

As pointed out in the introduction to this section, the method of user interface is the most hardware dependent of the six categories of design issues discussed. As technology advances, additional methods of interfacing with a computer system will likely be developed. While CAI technology has the flexibility to incorporate almost any specialized interface technique, this discussion will address only the more common methods--keyboard, mouse, light pen, infrared touch screen, interactive videodisk, and sound/voice.

1. Keyboard

This method is by far the most common method of user/computer interface and its popularity has naturally carried over into the field of CAI. The most obvious advantage to using the keyboard is an increase in the portability of the finished product since keyboards are an almost universal method of computer interface. In addition, the "standard" keyboard is familiar and can be readily adapted to by most students. Another major advantage of this method is that it allows for a much wider range of response to questions offered by the CAI program, and it facilitates creative student inputs.

The biggest drawback of using a keyboard as an interface is that, for the most part, CAI systems do not handle typing errors well. For example, if a system is expecting a student to type either "yes" or "no" and the student responds with a "yed", most CAI programs are not capable of recognizing this response and the program trips to an error routine. An effective method of limiting the effect of typos is to require only single character responses from the student where possible. This minimizes the opportunity for typos to occur. Another problem with keyboard usage is that moving the cursor around the screen to position a response or to select a displayed item is relatively slow and inefficient when compared to other methods. In addition, it should be pointed out that while most students are familiar wwith the keyboard, it has not yet been incorporated as a standard tool within the educational system to the extent that paper and pencil have.

2. Mouse

A mouse is a handheld tool with an embedded ball or LED light which, when moved across a mouse pad or a hard surface such as a table top, causes the cursor to be repositioned a proportional distance on the display screen. Use of the mouse is growing in popularity due to its capability to rapidly manipulate the cursor in any direction. It is especially useful when the user wishes to use the keyboard as little as possible. When incorporating the use of a mouse into the design of a CAI, it should be remembered that it is difficult to repeatedly shift from mouse to keyboard and back. Therefore, the total design of the CAI should be integrated into the use of a mouse. This entails such things as the modification of menu selections and less reliance on questions which require a character or textual response. Since the mouse is not a familiar tool to most students, it takes time to gain proficiency in its use. Therefore, the mouse should not be used in short CAI programs which will not be repeatedly used by a student unless the program is embedded in a sequential series of programs which utilize the mouse.

3. Light Pen

The light pen is a handheld device, resembling a pen, which is also used to facilitate the rapid movement of the cursor and to select areas of the display. The user touches the end of the light pen to the desired area of the screen. Pressure on the tip of the pen opens a gate which allows a small amount of light through to the screen. The screen senses this light and moves the cursor to the selected location. The advantage of the light pen over the mouse is that it gives the user "direct" control over positioning the cursor. It also may not require the extra space needed by a mouse, and is much simpler to learn to operate. Use of the light pen in conjunction with the keyboard is a relatively easy operation. The implication of these features in CAI design is that it is feasible to devote only certain portions of a program to use of the light pen and still retain the keyboard as a primary input device. This is especially significant because it facilitates the backfitting of current CAI software to take advantage of the features of a light pen where appropriate.

4. Infrared Touch Screen

This relatively recent development allows the user to select areas of the screen and to position the cursor by merely touching his finger to the desired area of the screen. The screen then senses the infrared radiation generated by the tip of the finger and moves the cursor to that location. The use of this method is analogous to the use of the light pen, substituting a human finger for the mechanical light pen. This sytem further eases operation in conjunction with a keyboard and, thus, can easily be incorporated into the design of a CAI system.

5. Interactive Videodisk

The interactive videodisk is primarily an output device which enables a much wider range of graphical and pictorial responses. The use of the videodisk can significantly enhance a CAI package which requires extensive use of visual material. To maximize the effectiveness of this method, detailed plans for its use need to be made during the design phase. The videodisk is well suited to the tutorial and simulation CAI strategies.

6. Sound/Voice

The generation of sound, an output mechanism, and the use of voice recognition, an input mechanism, can provide an added dimension to a CAI system. The use of various output sounds to enhance certain aspects of a CAI system is a relatively old and well understood technique. Feedback is one area in particular where the use of sound is appropriate. Today's computer systems can generate a wide array of sounds, from simple tonal renditions to the simulation of a human voice. Design issues relate to what type and where sound should be used. Other considerations include the environment the CAI will be utilized in. In a room with a number of terminals, the use of CAI with sound generation can be distracting to other students unless proper precautions are made (such as the use of headphones) [Ref. 63].

Voice recognition as an input device is a relatively new field which offers great promise to CAI designers. When this technology fully matures and becomes more widely available, the use of voice recognition in conjunction with the keyboard could become the primary method of CAI interface. Obvious advantages of voice recognition include the elimination of typos and, for long input strings, a faster response time is possible. In addition, the use of voice is a more natural means of communication. A number of technical problems must be resolved before this method can be fully incorporated into an effective CAI system, however.

	Advantages	Disadvantages
Keyboard	*Most common method of interface. *Increased portability of finished product. *Facilitates creative student inputs and allows for a wide range of response.	*Typos in student input must be considered in the design of the CAI. *Movement of cursor tends to be slow and cumbersome. *No single industry keyboard design.
Mouse	*Allows for the rapid manipulation and positioning of the cursor. *Can be used to minimize the use of the keyboard.	*Can be distracting and time consuming to repeatedly shift from the keyboard to the mouse and back. *Requires time to learn to use efficiently. *Difficult to effectively incorporate the use of a mouse into existing CAI systems.
Lightpen	*Gives the user direct control over positioning the cursor. *Simple to learn to operate. *Can be effectively backfitted to portions of existing CAI systems.	*Can require special CRT mounting techniques. *Almost always used in conjunction with another primary input device.

Figure 3. Advantages and Disadvantages of Various User Interface Methods

	Advantages	Disadvantages
Infrared Touch Screen	*Gives the user direct control over positioning the cursor. *Nothing to learn to operate. *Can easily be incorporated into CAI design.	*Requires special CRT hardware. *Not suitable in most cases as a stand-alone input device.
Interactive Video Disk	*Facilitates a much wider range of graphical and pictorial responses.	*Primarily an output only device. *Effective use requires detailed plans during the design phase.
Sound/Voice	*Particularly well suited to providing feedback. systems. *Elimination of typos. *Most natural means of communication.	*High cost associated with voice input *Voice recognition technology not fully matured.

Figure 3. Continued

V. LIFE CYCLE DEVELOPMENT METHOD FOR A DRILL AND PRACTICE TYPE CAI

This section will develop a standardized systematic methodology which can be applied to the development of a drill and practice type CAI. An earlier version of this methodology was successfully utilized by the author in the development of the portable microcomputer-based CAI package which is included in the annex of this thesis.

The developed methodology is divided into six distinct phases: (1) feasibility analysis; (2) initial design; (3) detailed design; (4) program development; (5) code and test; and (6) implementation and course integration. (The basic framework for this methodology is based on a composite of previous work in this area by Steinberg [Ref. 64], and Hartman [Ref. 65].) While in a macro sense, the execution of the phases is essentially sequential, many of the components of the individual phases can be completed simultaneously within each phase. This concept is illustrated in Figure 4, which outlines the various components of each phase and the execution sequence. In the application of this model to portable microcomputer-based systems, the initial design phase is critical. The design choices made during this phase must be carefully selected in order to ensure that the technology exists to implement these choices on a portable microcomputer system.

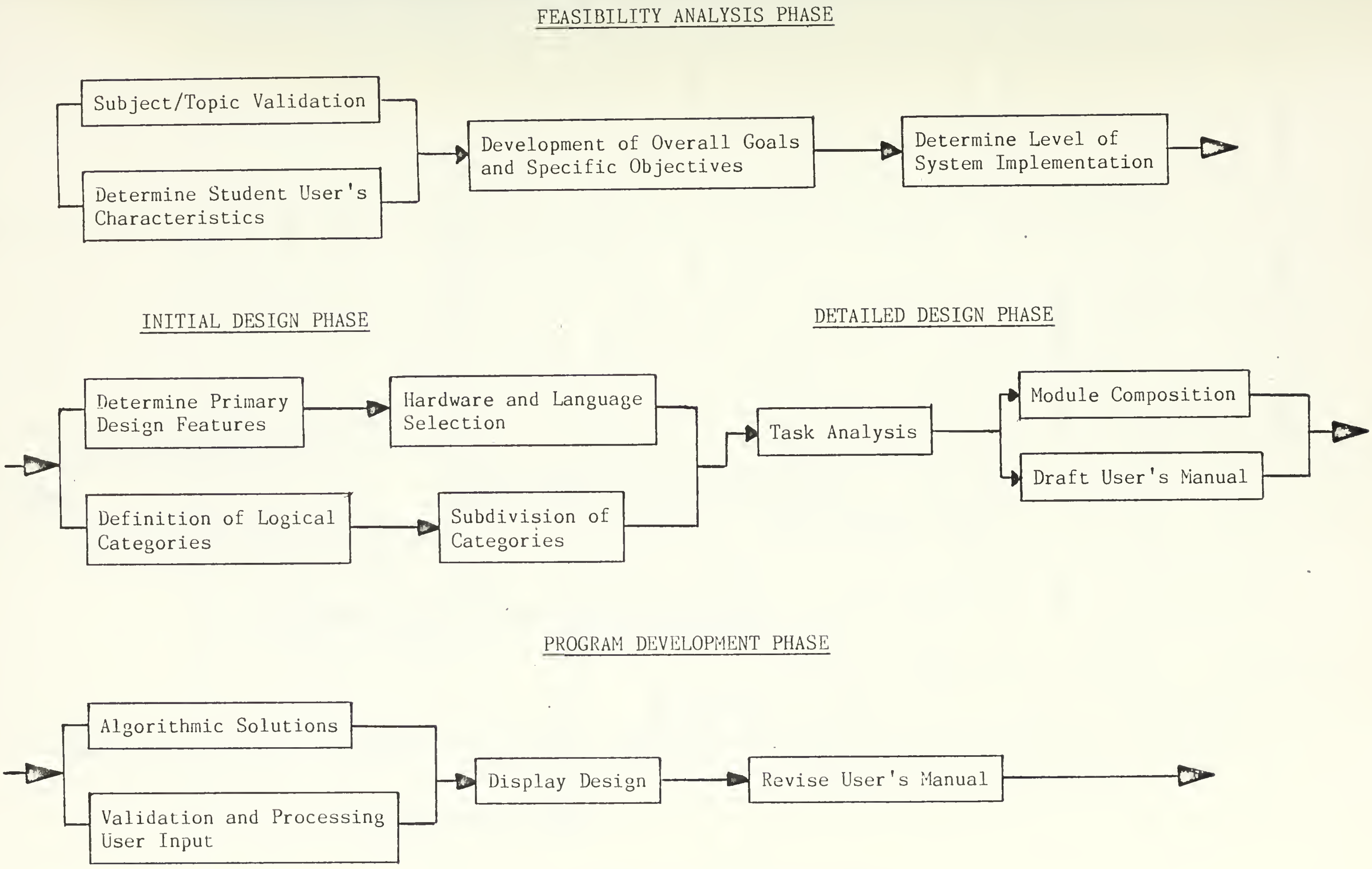

FEASIBILITY ANALYSIS PHASE
Subject/Topic Validation
Determine Student User's Characteristics
Development of Overall Goals and Specific Objectives
Determine Level of System Implementation
INITIAL DESIGN PHASE
DETAILED DESIGN PHASE
Determine Primary Design Features
Hardware and Language Selection
Definition of Logical Categories
Subdivision of Categories
Task Analysis
Module Composition
Draft User's Manual
PROGRAM DEVELOPMENT PHASE
Algorithmic Solutions
Validation and Processing User Input
Display Design
Revise User's Manual

Figure 4. Life Cycle Developmental Model

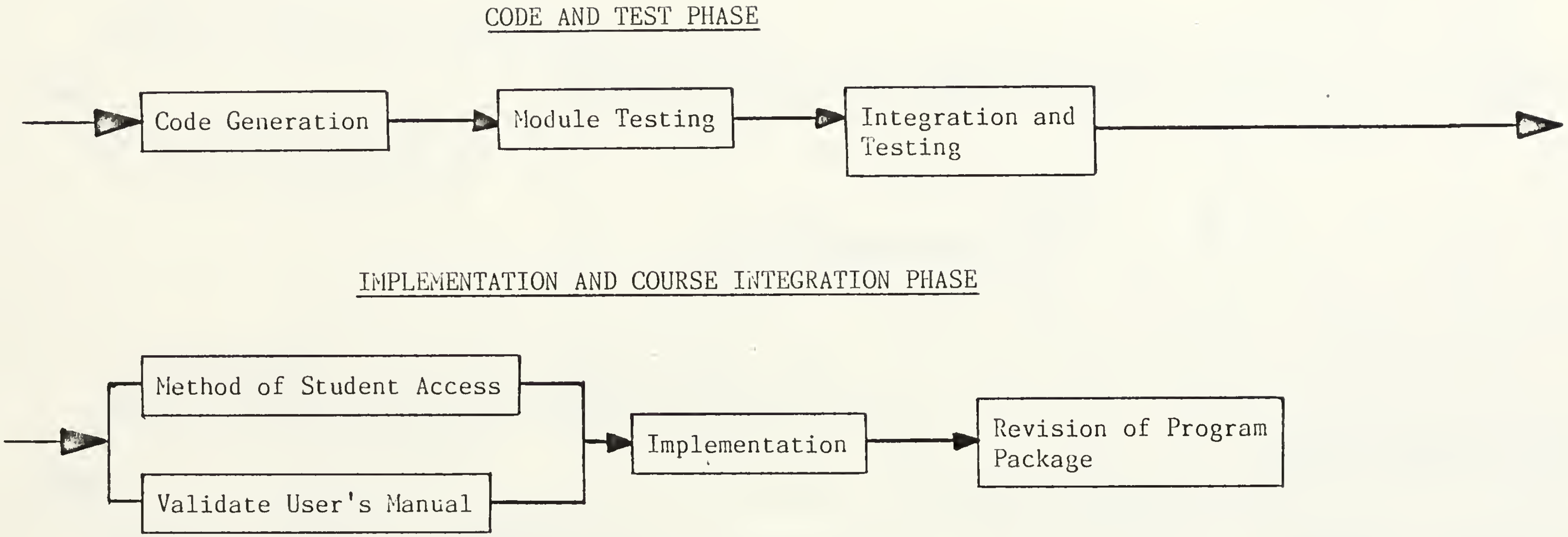


Figure 4 (Continued)

A. FEASIBILITY ANALYSIS PHASE

The basic objective of this phase is to ensure that the development of a CAI package to accomplish the desired application is feasible. A secondary goal is to ensure that all the decisions necessary to effect a smooth transition into the initial design phase are made. In order to accomplish these objectives, four distinct task areas have been defined: (1) subject/topic validation; (2) development of overall goals and objectives; (3) determine the student user's characteristics; and (4) determine the system level of implementation [Ref. 64]. Upon the successful completion of this phase, the output from these individual tasks will be used as guidelines during the subsequent phases. It is important to note that these guidelines are not "cast in stone" and are subject to modification as necessary. If, however, extensive modifications are made, it would be prudent to review each of the tasks in this phase to ensure the project remains viable.

1. Subject/Topic Validation [Ref.67]

Not all subjects/topics are condusive to the application of the drill and practice CAI strategy. In addition, some subjects may support this type of strategy but remain poor candidates for a CAI due to other reasons. An ideal subject/topic for the drill and practice CAI strategy will exhibit the following three characteristics:

a. Stable Curriculum Content

The composition of the course curriculum should be relatively stable and free from significant aberrations over time or between instructors. The content of a CAI system will parallel that of the selected course and cannot be rapidly and economically adjusted to sweeping changes in this content.

b. Problem and Procedure Oriented

The foundation of the drill and practice strategy is providing a student the opportunity to cycle through a number of similar processes in a rapid fashion. The subject/topic must contain problems or procedures which can be assimilated into this repetitive cycle in order to facilitate this strategy.

c. Sufficient Course Demand

The development and implementation of any CAI system is costly, both in terms of resources and time. While the drill and practice strategy has the advantage of having the lowest associated costs, it is still not a viable solution if the course for which it was developed is not offered on a recurring basis or has low student participation.

2. Development of Overall Goals and Specific Objectives

In the design of any system or organization, the need for well defined goals and objectives is imperative.

The development of such goals/objectives early on in the design of a CAI is equally important because it is their content which will determine if a drill and practice type strategy is appropriate. After verifying that the desired result can be achieved within the framework of this strategy, these goals and objectives can be used as a guide in the development and subsequent assessment of the program package. The development of these goals and objectives should be a joint effort involving both the instructor(s) and the potential CAI designers. An ongoing review should also be conducted to ensure that the goals remain valid and that the chosen objectives are contributing to the accomplishment of these goals. Major modification of these goals later in the lifecycle development can result in an invalidation of a project's feasibility.

3. Determine Student User's Characteristics

Knowledge of certain user characteristics such as educational level, previously acquired skills, maturity, and level of motivation are important in the subsequent design of a CAI. Obtaining a list of the required prerequisites and skills required for enrollment in the subject course is a good first step. To optimize various designer features, however, additional information about the potential users is necessary. Certain features can be incorporated into the design of a CAI to help compensate for low motivational users or educationally deprived users [Ref. 68; Ref. 69].

The addition of these features costs additional resources, however, so they should not be casually thrown into a design merely to meet all contingencies. When designing a system for implementation on portable microcomputers, it is particularly important that the user population be mature enough to properly handle and safeguard the issued microcomputer. (Otherwise, not only will the CAI be ineffective, but the tools to implement follow-on programs may be irreparably damaged.)

4. Determine System Level of Implementation

Information regarding the project goals and the user population's characteristics should be analyzed to detemine the ideal size of the computer system needed for implementation. While the statement of some goals might clearly support or even spell out what type of computer system is to be used, most will not. It is important to carefully integrate all the above information to ensure that certain goals or user characteristics do not require different or conflicting hardware configurations. If, for example, one of the goals is for the system to be capable of being implemented on a portable microcomputer, a careful analysis of other goals should be conducted to determine how much random access memory is needed, and if overlay techniques are going to be necessary. The advice of professionals who are familiar with the various hardware configurations under consideraton should be sought.

Remember that at this stage, a decision about the general system characteristics is all that is needed. Specification of exact hardware requirements is done during the next phase after the primary design features have been delineated.

B. INITIAL DESIGN PHASE

The basic objective of this phase is to decompose the valid subject/topic into coherent, logical categories of problems or procedures which can be subjected to a thorough task analysis in the next phase. Other objectives include the specificatiion of the desired primary design features, and hardware and program language specifications. To accomplish these objectives, four task areas have been defined: (1) definition of logical categories; (2) subdivision of each category into groups of problems with similar characteristics [Ref. 70]; (3) determine the primary design features; and (4) hardware and language selection. As illustrated in Figure 4, the accomplishment of these tasks is not purely sequential in nature. The selection of the primary design features and hardware/language selection can be accomplished in parallel with the definition and subdivision of the categories. In order for this phase to be completed successfully, a thorough knowledge of the subject/topic is necessary. If the potential designer of the CAI system lacks the detailed knowledge required, this phase should be conducted with the extensive outside assistance of a person, or group of persons, who have an adequate background in the subject. (Ideally, the designer of the CAI should be well versed in the specifics of the course to be automated. This is one of the best arguments for the development of a CAI system in-house where personnel with proven skills in the subject matter can be assigned CAI development tasks.)

1. Definition of Logical Categories

This task involves carefully reviewing the entire array of skills and procedures which are either currently being taught or are required to meet the goals developed in the previous phase. The first step is to develop an initial list or outline of these skills and procedures. Specific items on this list which cannot be adequately addressed by the drill and practice CAI strategy, or whose automation would not significantly contribute to the stated goals or objectives, should be crossed out. The items which remain represent a complete listing of the type of procedures or skills which will be addressed by the developed CAI system. An initial attempt should then be made to define a series of broad categories of items from the list. Ideally, these categories will represent a logical sorting of the items, based on an in-depth knowledge of the subject matter. For example, the calculation of the mean of a data set should not be placed in a category titled, "Inferential Statistics" but would be more appropriate placed under "Descriptive Statistics". The resulting logical categories should be reviewed to ensure they are defined in a manner consistent with the program's goals and objectives.

2. Subdivision of Categories Into Smaller Groups

In order to simplify the detailed task analysis which will be conducted during the next phase, each of the categories created above should be subdivided into individual groups of problems. Each of these groups should exhibit similar computational and/or theoretical characteristics. The separation of problems into groups in this manner will ease the development of solution algorithms and serve as the initial step in the definition of distinct program modules. The programming of problems which are solved using similar techniques can often be accomplished with the development of one common use algorithm. Groups of problems which draw on the same theorems, axioms or postulates for solution will, in many cases, require certain standard values which can be stored once in a program and accessed by various problem solving algorithms when necessary. This computational and theoretical grouping reduces program redundancy and usually results in the saving of memory space. This procedure suggests that the result of defining a small number of large groups will be a significant reduction in the number of required solution algorithms. Rigid adherence to such a procedure can, however, unnecessarily complicate a program and extend development time. Algorithms which are designed to solve an excessively wide range of problems tend to be more complex and time consuming to develop, and especially difficult to adequately test [Ref. 71]. Therefore, care should be taken to ensure that each grouping has well defined boundaries that effectively narrow the range of problems to be addressed to a manageable level. Intelligent development of these groupings can result in a significant reduction in the time and effort required for overall project completion.

3. Determine Primary Design Features

This is the most critical task to be completed during this phase. It involves making decisions about CAI design issues such as the method to be used to effect user interface, certain enhanced screen design issues, and the desired level of guidance, feedback, and HELP. The decisions made must be compatible with both the goals and objectives of the program and the general computer system characteristics which were developed during the previous phase. Due to the importance of decisions about these CAI design issues, it is wise to involve both the potential course instructor(s) and select student users.

Some of the more common user interface methods include the use of a keyboard, mouse or lightpen. (The design issues related to each of these interface methods were discussed previously in this thesis.) Due to the unique programming requirements required by certain interface methods, it is critical that a selection be made prior to advancing to the detailed design phase. The retrofitting of a different method of interface to an already completed CAI can involve a significant amount of reprogramming and, depending upon the new interface method, may not be feasible within the framework of the existing program. The selected interface method should be either familiar or easily learned by the user population, and should provide a method to rapidly respond to the type of queries anticipated for the developed CAI.

The actual layout of individual screen displays and the associated design issues takes place during subsequent phases. Enhanced screen display design features such as the use of color and graphics should be addressed during this phase before actual hardware selection. It is again emphasized that color or graphics should not be introduced to a CAI unless they contribute to the overall goals of the program.

Decisions made during this phase regarding the levels of guidance, feedback, and HELP directly impact on the method and complexity of the programming task. In general, the higher the desired level in any of these three categories, the more complex the resulting programming effort will be. In CAI systems, the issues of guidance, feedback, and HELP permeate through the development of the entire program. Therefore, it is important that these issues be decided upon early, and that the resulting decisions adequately address the learning needs of user population.

4. Hardware and Language Selection

This task involves the selection of the proper hardware and programming language which will be used to implement the developed CAI. While hardware selection and programming language selection can be discussed as separate issues, the interrelationships between the two (particularly in microcomputer systems) require that this selection process be a combined procedure. The actual selection should be made from within the framework of the general system characteristics developed during the Feasibility Analysis phase, and should incorporate the capability to support the desired design features outlined above.

An initial estimation as to how many complete systems and the number of individual workstations required should also be made at this point. The quantity of both can be expected to vary with the nature of the CAI, how it is used, and the cost of the individual components. In most cases, this estimate can readily be modified at any future point in the development without seriously affecting CAI programming. Exceptions to this include cases where special programming techniques may be required to support multiple users in a time sharing environment.

Since it may not be possible to obtain a cost effective hardware/programming language package to achieve all the desired design features, it may be necessary to go back and reassess which design features are required. This process should continue in an iterative fashion until a satisfactory match has been made. In all cases, the various planned requirements and design features should be reconciled with the selected hardware/programming language package before advancing to the next phase.

C. DETAILED DESIGN PHASE

The primary objective of this phase is to convert the initial concepts and design features developed in the previous phase into a detailed set of plans from which appropriate algorithms can be drafted. This phase combines the use of a bottom up approach to analyze each distinct problem type with a top down view of overall module composition and the development of the proper control flow links between the modules. Three separate task areas have been defined for this phase: (1) detailed task analysis for distinct problem types within each group; (2) individual module composition and flow between modules; and (3) initial draft of user's manual. The completion of these tasks will also provide excellent documentation for use in program testing and maintenance. This phase is directed primarily at the development of medium to large scale applications of the drill and practice strategy. In these cases, it is not practical to design a large number of algorithms directly from the initial concepts and a list of desired design features. In smaller scale or relatively simplistic applications, it may be possible to skip this phase and proceed to the Program Development phase.

1. Detailed Task Analysis

A detailed task analysis designed to document the solution process involved in working through a problem is conducted for each problem type within each group. This analysis should reveal all of the intermediate steps and various solution paths for each problem type. The results of this analysis will form the basis for the algorithmic solutions to be developed in the next phase. In order to adequately support the development of these algorithmic solutions, the analysis must provide a complete definition of the primary and secondary logical solution paths, a detailed map of all user input requirements and their sequence, and a draft of the appropriate HELP display for each problem type.

In most problems or processes, there are a number of valid methods available to reach a solution. Each of these methods needs to be explored and expanded into a step-by-step procedure which represents a distinct path. After all feasible paths have been defined (assuming a finite number of feasible paths are available), one or two should be designated as "primary paths" with the remaining paths termed "secondary paths". It is the primary paths which, as a minimum, will be developed into algorithms for implementation within the CAI program. The number of secondary paths which are eventually incorporated into the program will vary with the complexity required for their implementation and the stated objectives of the concerned application program. The value of having a complete set of paths defined is that it precludes the possibility of overlooking a potential primary path and it provides the documentation to support a rapid expansion in the number of available solution paths should the need arise.

Associated with each of the above solution paths are a series of user inputs. These inputs include such things as problem parameters, raw data, and responses to program generated questions. In some cases, a precise ordering of such inputs is required to support the sequential nature of the step-by-step procedures outlined in the solution paths. A detailed map of the user input requirements associated with each path should be developed, along with any required sequencing information. The map of user inputs defined for the primary path should be expanded upon to include convenient points where the CAI program could reinforce the learning process through appropriate interaction with the user. This map will be used extensively in developing the algorithms which affect the interface between the CAI program and the student.

In conjunction with individual problem analysis and the mapping of the input process, a draft of the contents of any potential HELP displays for each problem type should be developed. HELP displays should be generic to the problem type (not path dependent) if possible; otherwise, the incorporation of additional paths or minor changes in the selected paths will also necessitate changes in the HELP displays. The level of detail of these generic HELP displays will be dictated by the decisions made during the Initial Design phase. Some CAI systems are designed with little on-line HELP and instead rely on extensive verbal prompts and menus. In such cases, the time saved during the completion of simple (or no) HELP displays will typically be consumed in designing the menus and prompts.

2. Module Composition and Inter-modular Flow

The various solution paths for all problem types within each group should be studied to determine which paths can be combined into a single module. In some cases, a group of problems may be suited for decomposition into one or two modules. In most cases, however, in order to maintain an adequate degree of module cohesion, a number of separate modules are required. (Since the problems were previously grouped into logically similar types, it is unlikely that a single module will need to contain member paths from two different groups.) The standard programming practices of minimizing intermodular coupling and maximizing module cohesion apply. When viewed outside the context of this development model, the process of grouping a long list of procedures or paths into separate modules can be an overwhelming task. The provided model eases this process by distributing many of the tasks normally associated with module development over two phases, starting with the definition of logical categories in the Initial Design phase and ending with a set of distinct modules in the Detailed Design phase.

After the composition of the various modules has been developed, a top down approach is used to determine the proper ordering of the modules and to provide the mechanism which will control the flow between modules. The results of this approach should provide the basis for the development of an algorithm which can execute a series of physical modules in a logical fashion easily followed by the user. Adjustments to module composition may be necessary to accommodate the proper logical sequencing of program flow. As part of this process, a common method to handle certain classes of errors should be developed. In a CAI, it is assumed that students all make logical errors in responding to system generated queries. Each individual module should have internal mechanisms to handle logical errors. (This minimizes delay in providing feedback.) Other classes of errors, such as control flow and interface errors, are handled more efficiently with a common error routine.

3. Initial Draft of User's Manual

The development of an initial draft of a hardcopy user's manual should be conducted in parallel with the detailed task analysis and the decomposition of the problem groups into modules. Drafting a rough of a user's manual at this early stage enables documentation of user procedures to correspond more closely to actual program operation [Ref. 72]. The final version of the user's manual will typically be evolutionary in development and this model allows for ample flexibility in adjusting the content and format of the manual in later phases. A wide variety of types of users' manuals are possible and the detail required for a particular application will be directly related to the method of user interface and level of guidance specified during the Initial Design phase.

D. PROGRAM DEVELOPMENT PHASE

This phase is primarily concerned with the development of detailed algorithms to implement the various problem solutations, HELP displays, control features and error handling mechanisms defined in the previous section. The design of individual screen and menu displays is also accomplished during this phase. The output from this phase should easily accommodate conversion into executable code in the language selected during the Initial Design phase. This phase includes four specific task areas: (1) develop algorithmic solutions to each specific problem type; (2) develop algorithms to validate and process user input; (3) design individual screen and menu displays; and (4) review and revise user's manual. The development of the various algorithms should be easily accomplished through use of the "solution paths" and the map of user input requirements which were completed as part of the previous phase. The most important and in most cases the most time consuming task to be completed during this phase is the design of the individual screen and menu displays. A number of issues dealing with basic screen design must be considered in order to obtain an optimum style which effectively integrates the display capabilities of the chosen system with the goals and objectives of the CAI program.

1. Development of Algorithmic Solutions

As mentioned above, the tools created during the previous phase should be used as blueprints in the development of algorithmic solutions to each specific problem type. It is important to remember, however, that the "solution paths" developed in the previous phase represent logical routes, and are not necessarily the most efficient method to obtain a computer derived solution. These paths can, however, be used as a starting point in the development of computer efficient algorithmic solutions.

Whatever the form of the final version of the developed algorithms, all interaction with the user should conform to the steps detailed in the "logical solution paths", and the internal mechanics used to derive the computer solution should be transparent.

A modular approach should be taken in the development of the algorithms. The definition of module composition and functions has already been delineated in the previous phase. The algorithms which will be involved in implementing the required functions of a single module should be developed together as a unit. This practice helps eliminate execution conflicts within a module.

2. Validation and Processing User Input

A series of algorithms to validate and process user input should be developed based on the map of user input requirements developed during the Detailed Design phase. Other factors which should drive the design of these algorithms include the desired level of user guidance and feedback which were selected as part of the Initial Design phase. Each student input needs to be validated before it is passed to a module for use in calculations or comparisons. Provision must be made to provide feedback to the user based on the results of this validation process. In addition, feedback mechanisms must be developed which will respond to valid but incorrect user inputs. For example, if a student is asked to calculate and input the standard deviation of a data set, the CAI program should be capable of responding to invalid inputs (e.g., a negative standard deviation), as well as valid but incorrect values. In this situation, the feedback provided when a negative value is input would typically advise the student that the standard deviation cannot be less than zero and request another input. After the student provides a valid input, the program should check the solution for accuracy and advise the student accordingly. If the student's solution is correct, a brief statement complementing the student on his or her statistical skills may be appropriate. If an incorrect answer is input, the program should, at a minimum, alert the student to the error and perhaps provide another opportunity to input the correct solution. The range and the depth of this feedback will depend on the application and on the selected level of feedback.

3. Screen and Menu Display Design

Prior to the actual start of design work, a significant amount of effort should be devoted to analyzing the various screen design issues discussed in the previous section of this thesis. The results of this analysis should be used to develop an overall screen design style. This style can include a variety of combinations to accommodate various display requirements within the program, but the use of a particular screen design should be consistent within similar applications. This consistency better enables a user to gain proficiency in executing the program. The design of screen and, particularly, menu displays will be heavily influenced by the choice of user interface method and the display device. A distinctly different menu display is necessary, for example, to optimize the use of a mouse vice a keyboard as an interface device. Likewise, if the potential display device operates in 40 column format, the various screen and menu displays should be tailored to this format. To improve the portability and to lengthen the viable lifespan of the finished product, the displays should be designed so that they can easily accommodate, and take advantage of, anticipated future hardware upgrades.

One of the real challenges associated with this task is to decide exactly what type of screen design style should be used in educational interaction with the student. Methods commonly used in a drill and practice type CAI include the ever popular multiple choice selections and the use of a series of well defined questions in a stepwise fashion. Choosing the optimal method is based on a number of diverse factors ranging from learning theory to hardware restrictions, and is beyond the scope of this thesis.

4. Revise User's Manuals

Typically, during the algorithmic and screen design process, a number of minor modifications to the projected structure of the program will occur. Any of these changes which affect the user's view of the program should be reflected in the user's manual. This task is usually easy to accomplish in conjunction with the other tasks in this phase, but if overlooked can render large sections of the user's manual obsolete. It is not uncommon to attempt to delay making revisions to the user's manual until after the total software package is complete. This approach can result in a considerable time lag between the incorporation of changes in the software package and the revision of the user's manual to reflect these changes. This time lag increases the chances that pertinent revisions to the user's manual willl be given short shrift or completely overlooked.

E. CODE AND TEST PHASE

This phase concentrates on the development and testing of executable program code derived from the algorithms and display designs written in the previous phase. This phase steps through the coding of the individual modules, the corresponding testing of these modules, program module integration, and final testing of the program as a complete system. The programming language to be used in implementing the code was selected earlier during the Initial Design phase. Obviously, the details of the coding of the algorithms will differ depending upon the characteristics of the programming language. In general, however, this model recommends the adoption of a bottom up strategy in the coding of the program in order to facilitate the separate testing of each module before it is integrated with the main program [Ref. 73]. In the development of any large software product, "bugs" are inevitable and will continue to crop up throughout the lifecycle of the product [Ref. 74]. In a CAI application where the user population will mainly consist of students, it is particularly important to try and eliminate as many problems as possible through testing. For this reason, special emphasis is placed on various methods of testing the coded program.

1. Individual Module Code Generation

Conversion of the previously derived algorithms into executable code should be done on a module to module basis in a bottom up fashion. In the selection of variables and parameters, special emphasis should be made to ensure that there are no conflicts within a module. Maximum use of local variables is recommended to minimize unexpected and hard to trace errors elsewhere in the program [Ref. 75]. If the selected language does not support local variables (e.g., BASIC), then an added measure of caution is needed to ensure no conflicts arise. The use of a high level of feedback or providing extensive user guidance in a CAI program usually results in the generation of a good deal of textual material. Spelling errors or other gross grammatical errors in this text can tarnish the reputation of the best CAI programs. By nature, CAI programs interact extensively with the user. To implement this interaction, the programmer must be aware of the various input/output capabilities supported by the selected language.

Informal checking and testing of various components of a module should be done concurrently with module development. After completion of the module, it should be subjected to more thorough formalized testing procedures.

2. Module Testing

When using the bottom up strategy for coding a program, a dummy driver will be required to effectively test a module. This dummy driver should be simple in design and limited to the input/output requirements of the module to be tested. After this dummy driver has been constructed, the formal testing process can continue. This model recommends the use of both whitebox and blackbox testing techniques. A brief description of each technique follows. For a more detailed treatment of these and other testing techniques, the reader is referred to [Ref. 76].

--Whitebox Testing. Whitebox testing (also known as logical or internal testing) is based on the internal construction of the module. Various logical paths in the module are tested by providing tailored input values. It is generally not possible to test every available path in a module. A well designed test, however, will exercise the primary paths through the module.

--Blackbox Testing. Blackbox testing (also known as input/output testing or data driven testing) is based on testing the various software interfaces within the module. Test data is generated for input and the corresponding output is inspected for accuracy.

After modification of a module in response to discovered errors is complete, the new version should be retested to ensure that the "fixes" did not introduce more errors. When a complete battery of tests has been completed revealing no additional errors, coding of the next module can begin. This process continues in an iterative fashion until all of the individual modules have been coded and successfully tested.

3. Program Module Integration and Testing

This task involves integrating all of the individual modules into a complete system. The coding of a driver routine and any ancillary routines must be accomplished before the program can function as a system. The guidelines developed in the previous two phases which pertain to program flow should be used to code the driver routine. Actual testing of the complete system can begin when this coding process has been completed. Testing of the system as a whole is conducted using basically the same techniques used in module testing.

When an error is detected during system testing, the first place to check in searching for the problem is the driver routine. If the driver routine is not to blame, then the problem is most likely caused by interface procedures between modules. Although the procedures of bottom up module/system testing will isolate most problems to one of the areas specified above, in reality the problem could be located at any point in the program. This is one reason why "debugging" is often referred to as an art rather than a science.

F. IMPLEMENTATION AND COURSE INTEGRATION PHASE

This final phase deals with the implementation of the completed CAI program and its integration into a course of instruction. A great deal of discussion in the literature is devoted to a seemingly endless list of general CAI implementation and integration issues. This model, however, is focused only on the development of a drill and practice type CAI which performs a relatively well defined function within a given course. Since the overall purpose of a drill and practice CAI package has been "predefined", the remaining issues associated with its implementation are relatively straightforward. Many of these issues should have already been addressed during the Feasibility Analysis phase when the goals and objectives for the program were defined. Furthermore, methods for approaching other ad hoc issues which may arise during this phase can usually be readily determined through examination of these goals and objectives. The number and kind of tasks which can be encountered when executing this phase will depend on the particular application, and a complete enumeration of these tasks is beyond the scope of this model. Three basic tasks which are common to most applications merit a brief discussion: (1) determine method of student access; (2) validate user's manual; and (3) revise program package based on user feedback.

1. Method of Student Access

Now that the CAI program package has been prepared and is ready to be implemented, a method of allowing students adequate access to the package and the associated computer hardware is needed. This access scheme will vary depending upon the type of computer system involved. On medium to large size computer systems, the primary issues revolve around the number of available terminal hours. In order to be effective, a drill and practice CAI must be used on a regular basis. Depending on the current level of usage of the computer system, providing adequate access to terminals at reasonable hours may be a problem. On microcomputer-based systems, the availability of microcomputers can cause similar problems. The increasing proliferation of personal microcomputers, however, is contributing to the elimination of this problem. Individual student use of personal computers raises the issue of how the CAI package can be distributed to each student. Common methods include providing each student individual disk or tape copies of the program, and downloading the program from a larger central computer via modem links. Each method has distinct advantages and disadvantages.

2. Validate User's Manual

Prior to actual student use of the developed package, the user's manual should be validated for accuracy and applicability. This process is best performed through actual interaction with the CAI package under conditions similar to those that the students can expect to encounter. After the manual has been properly validated, it can be turned over for student use. Further modification of the manual based on the experiences of the first group of student users is likely to be required.

3. Revision of Program Package

When the CAI package is initially implemented, a series of (hopefully) minor problems, which were overlooked during product testing, are likely to manifest themselves. In order to correct these problems, a well defined set of procedures for obtaining and acting upon user feedback must be developed. In addition to pointing out problems, students often can contribute excellent ideas for enhancing the CAI package. Typical methods of facilitating this communication between the students and the CAI designer include face-to-face discussions and survey/questionnaire forms.

VI. DESCRIPTION OF THE DEVELOPED CAI PACKAGE

This section will be devoted to providing a description of the drill and practice type CAI which was developed as part of this thesis. This package, which was designed to assist in the instruction of a graduate level introductory statistics course, was specifically tailored for implementation on the Radio Shack Model 100 portable microcomputer. (A description for the RS Model 100 can be found in Appendix C.) The use of a truly portable microcomputer offers significant advantages in availability and cost which are not possible with larger systems. Greater quantities of these highly capable devices can be purchased, thereby permitting individual issue to each student user. The portability of the RS Model 100 allows the completed software package to be used in almost any environment, at times most convenient to the student.

A complete set of program listings and the associated user's guide are provided in the annexes of this thesis. The user's guide provides, among other things, example runs of each type of statistical procedure addressed by the package. Perusal of these example runs is a good way to become familiar with the various screen displays and functions available within the package.

A. OBJECTIVES

The primary objective of the developed package is to provide a tool which will enable students enrolled in an introductory statistics course to make more effective use of time spent working on homework and example problems. In addition, it is hoped that this package will demonstrate some of the unique capabilities offered by portable microcomputers in the area of computer assisted instruction.

B. SCOPE

This package consists of a series of eight modules which provide assistance in the areas of descriptive and inferential statistics, the application of various distributions and random number generation, and simple and multiple regression. A complete listing of the contents of the eight program modules is provided below.

Module 1

Permutations
Combinations
Bayes' Theorem

Module 2

Finite Populations
Exponential
Normal
Uniform

Module 3 (Applications of Various Distributions)

Binomial Distributions
Geometric Distributions
Poisson Distributions
Exponential Distributions

Module 4

Central Limit Theorem
Applications of the Normal Distribution

Module 5

Maximum Error of an Estimate
Determination of a Minimum Sample Size to Achieve a Given Level of Error
Confidence Interval About the Mean
Hypothesis Testing of the Mean

Module 6

Confidence Interval About the Standard Deviation
Hypothesis Testing of the Standard Deviation

Module 7 (Linear Regression and Analysis)

Hypothesis Testing of Alpha and Beta
Confidence Interval for Alpha or Beta
Confidence Interval for the Mean of Y at X Naught
Limits of Predictions
Estimates of Y When Given X

Module 8 (Multiple Regression)

Significance Testing for R
Coefficient of Determination
Estimates of Y When Given X1, X2

C. DESIGN METHODOLOGY AND CONSIDERATIONS

The framework for the overall design of the package was provided by an earlier version of the lifecycle development model outlined in the previous section. The package is currently in the early stages of the Implementation and Course Integration Phase. At this point, approximately 690 man-hours, spread over a six month period, have been invested in its development. The distribution of development time among the various phases is depicted in Figure 5.

The content and flow of the individual modules of the program are based on the assumption that students will learn the various statistical techniques in the traditional classroom setting and the program will only be used to reinforce this knowledge through repetition. In an effort to optimize the benefits derived from the use of the computer, the modules are designed to present logical solution steps to each problem which mirror the steps necessary to solve the same problem by hand without the aid of a computer. In addition, the student is required to perform some simple calculations and table look-ups. All data input is validated, student calculations and table look-ups are checked for accuracy, and instant feedback is provided.

The program is designed as a partially menu driven system and makes extensive use of detailed user prompts. The use of the package does not require any special programming or computer skills and no memorization of confusing acronyms (beyond those commonly used in statistics) or special computer commands is necessary. The above features enable a student to effectively execute any portion of the program without any prior special preparation. This allows the instructor to devote his or her efforts totally to statistical matters and not waste

Phase	Man-hours	% of Dev. Time
Feasibility Analysis Phase	32	5%
Initial Design Phase	87	12%
Detailed Design Phase	150	22%
Program Development Phase	225	33%
Code and Test Phase	176	25%
Implementation and Course Integration Phase	20	3%
TOTAL	690	100%

Figure 5. Distribution of Development Time by Phase

precious class time reviewing special computer procedures which contribute nothing to the understanding of the subject.

The desired implementation of this package on a portable microcomputer required the incorporation of a number of special design considerations. A significantly smaller amount of available random access memory space and limited options in the development of screen displays are the major restrictions imposed by current generation portable microcomputers. The version of RS Model 100 used in implementing this package is typical of the "low end" portables and has a RAM capacity of 24k and an eight line by forty character liquid crystal display. To cope with the restrictions in memory capacity, the package was developed in a series of stand-alone modules. The complete package is segmented into eight modules which range in size from 9k to 15k. The various menus and screen displays were individually tailored to the Model 100's screen.

Initial modules were developed with a limited HELP facility. This feature was soon abandoned when it became apparent that the amount of memory and computer overhead it consumed could not be justified by the severely limited amount of information it provided. Input error correction and control is provided by the internal algorithms throughout each module. The package was subjected to a series of tests to eliminate other system errors. Nevertheless, in a software package of this size, a significant number of errors may "slip through" the testing process. To deal with these "undetected" errors, a simple error trap routine which terminates processing normally has been incorporated into each module.

D. DESIGN ISSUES ENCOUNTERED

The early selection of the RS Model 100 portable microcomputer as the implementation device effectively narrowed the range of the required design decisions during the development of the included package. Such issues as language selection, method of user interface, and many of the screen design issues were fixed by hardware restrictions. BASIC is the only programming language supported by the Model 100, and a keyboard is used with an eight line by forty character liquid crystal display to effect user interface. The most troublesome design issue was determining the optimum physical layout of each program display within the restricted screen size of the Model 100. In many instances, the inability to anticipate the length of the user's response to queries generated by the program made it impossible to design a display of fixed size. Consequently, certain screen displays may change in appearance depending upon the user's input data.

The limited amount of memory available on the Model 100 also led to the abandonment of an online HELP capability. To compensate for the lack on an online HELP facility, the level of user guidance had to be increased. In retrospect, this may not have been a wise decision as it led to excessively long narrative displays to explain each problem solution step. A better solution may be to incorporate a limited online HELP facility which acts to key the user to pertinent sections of a hardcopy user's manual.

E. CONSIDERATIONS IN USING THE DESIGNED PACKAGE

The developed CAI package is tailored to provide assistance to the student in working through a series of example problems within select topical areas of statistical analysis. The program is designed as an adjunct to the traditional classroom instructor and shall only be used after the proper subject material has been covered by the instructor. The sectioning of the program into distinct modules which group similar statistical concepts allows the instructor maximum flexibility in sequencing the course. While the modules are numbered one through eight, there is no requirement, or even any advantage, to utilizing the modules in this order. Due to memory space limitations on the RS Model 100, the maximum number of modules which can simultaneously reside on the computer is restricted to one or two. With this in mind, to allow for optimal student use of the program, the instructor should refrain from a singular assignment of a group of example problems which would require the use of more than two modules.

Ideally, the course should be presented in chunks which roughly correspond to the modular breakdown of the program. This should be fairly easy to accomplish as the composition of each module is primarily based on a logical partitioning of the subject matter. Following the presentation of a new statistical concept, the student should, at his leisure, load the corresponding module into the RS Model 100 (following the instructions in the User's Guide in Appendix A), and work through all assigned examples. The structure of each module was carefully designed in an effort to strike an appropriate balance between those calculations done by the student and those completed by the computer. It is this author's opinion that the student should be required to complete all simple calculations and table look-ups, with only the most extensive and tedious calculations being accomplished solely by the program. Nonetheless, the use of the program should significantly accelereate the completion of assigned homework, and the student is encouraged to use the additional time by working through other examples of his or her choice. The program can handle examples from any source, providing they can be structured into a format which will provide the student with all the necessary input parameters.

When using the program modules to work examples, the student should carefully note the presented solution process. The program modules, almost without exception, present only the "primary solution path" for each problem type. The student should rapidly catch on to these simple solution steps and be able to apply them without the aid of the computer. In addition, by using the problem solving skills learned through the use of the program as a foundation, the student should branch out and explore alternate paths on his or her own initiative.

F. LESSONS LEARNED

Throughout the development of this CAI package, a series of minor problems and issues had to be dealt with. In the process of grappling with these problems, a number of "lessons" were learned. Several of these lessons are presented here to provide the potential CAI designer some insight into the type of ancillary issues and problems likely to be encountered.

Perhaps the most valuable lesson learned is that the design of each program module for a CAI system requires a detailed knowledge of the subject matter and a conscious effort to determine how the topics within each subject are best taught. Attempts to design program modules based only on a working knowledge of each topic usually proved to be premature and counterproductive. The program designer(s) must devote a significant amount of time to obtaining the required level of expertise within the topic area prior to module design and development.

Once the hardware selection has been completed during the Initial Design Phase, the program designer(s) must become cognizant of not only the limitations and basic capabilities of the selected device, but also of any special capabilities. For example, certain special capabilities of the Model 100 (e.g., reverse video) were not discovered by the author until the Code and Test Phase, too late to be easily integrated into the program package.

The final point to be made addresses the primary storage device used by the designer in developing the system. During the development of this package, a cassette tape provided the only means of storing partially completed programs. This method proved to be awkward and time consuming. If at all possible, an alternate means of primary storage (such as disk) should be used in the development of the programming package regardless of the final hardware configuration which will be provided to the student users.

VII. FUTURE OF PORTABLE MICROCOMPUTER-BASED CAI

The future development of educational programs for use with portable microcomputers is tied directly to advances in the field of general CAI design and to continuing hardware technological improvements. CAI design should be profoundly affected by developments in artificial intelligence and cognitive psychology [Ref. 77]. Likewise, CAI should benefit as various problems associated with speech processing are solved [Ref. 78]. The incorporation of any of these developing technologies into portable microcomputers, however, is, in my opinion, a long way down the road due to the extensive hardware support I envision being required to support these techniques.

Nevertheless, the capabilities of portable microcomputers will undoubtedly continue to expand as evidenced by the newer models available today which exhibit many characteristics previously only available on "fixed site" microcomputers. These new capabilities should allow for the execution on portable microcomputers of educational programs previously developed for use only on the larger micros. Portable microcomputer technology is expanding so fast that, in the near future, with the exception of some of the advanced techniques mentioned in the previous paragraph, restrictions in hardware capabilities will not seriously hamper the development of complex CAI programs. The major drawback, as I see it, will be the cost of these advanced portable microcomputers.

The purchase price of the newer portables may be reduced somewhat, but this reduction is likely to be insignificant in the near term. The benefits associated with portable microcomputers in education cannot be fully realized without individual issue of the devices. Therefore, unless the costs of these advance models is reduced substantially, most educational institutions probably will not be able to afford the necessary large scale purchases. In the interim, however, the use of the cheaper portables such as the RS Model 100 can provide a much needed new diminsion to educational computer usage.

This thesis has reviewed the evolution of CAI and provided a working example of what I believe is the next logical step in CAI development--the incorporation of the portable microcomputer. The restrictions imposed by the current generation of affordable portables makes the "drill and practice" strategy particularly attractive. As such, a developmental model for the implementation of this type of strategy was outlined. A version of this model was used to develop a CAI program to aid in the instruction of introductory statistics. The number of applications for productive educational use of portable microcomputers is limited only by one's imagination. The development of quality software is the only real stumbling block, and with adequate motivation, this can be overcome.

APPENDIX A

PROGRAM USER'S GUIDE

By

LT Randall L. Henderson

United States Navy

Naval Postgraduate School

Monterey, California

1985

TABLE OF CONTENTS

Part I: INTRODUCTION 114

Program Objectives 114

Purpose of This Guide 115

Part II: GETTING STARTED 116

Downloading Executable Programs Using the Model 100's Internal Modem....................................117

Loading the Programs Via Cassette Recorder 122

Part III: EXAMPLE PROGRAM RUNS 125

Format of the Example Program Runs 126

Example Program Run Index 128

Formulas and Table Descriptions 210

References 281

Part I: INTRODUCTION

Program Objectives

These programs were designed to facilitate the drill and practice of select statistical methods. The logical flow of the programs matches the procedures which would normally be used to solve each type of problem without the aid of the computer. It is assumed that the user will learn the course material via the traditional classroom lecture. The value of the programs is in providing the student with a "boiler plate" approach tailored to solve each type of covered problem. The user can expand upon these basic techniques, learned through repetitive use of the program modules, on his own. The programs interactively guide the student through the use of prompts which request data about the problem and require the student to perform simple calculations and table look-ups. All data input is validated, student calculations and table look-ups are checked for accuracy, and instant feedback is provided. The sequential flow and the wording of the prompts has been designed to allow the student to concentrate on mastering the subject matter, not the program modules. In order to optimize the allocation of the student's time, complex calculations are carried out by the computer. This combined approach provides the student the time needed to explore a wider range of problems of the same type. Through interaction with the program, the student should gain a first-hand understanding of the slight variations possible when working through a set of problems which apply essentially the same statistical methods and theories.

Purpose of This Guide

The objective of this guide is to provide the user with the basic information needed to load and begin execution of the program modules. The guide is not intended to be an exhaustive summary of each module, nor is any effort made to describe all the possible screen displays and paths available in each module. The programs are partially menu driven, and make extensive use of self-explanatory prompts and amplifying statements where applicable. This extensive "internal" documentation eliminates the need for frequent reference to offline "external" documentation.

Part II of this guide contains, or makes reference to, the information necessary to master the mechanics of loading the program modules for execution. The last section of this guide, Part III, is devoted to examples which provide a taste of each of the modules operates.

Part II: GETTING STARTED

The first step is to familiarize yourself with some of the basic features of the Radio Shack (RS) Model 100. A review of the manufacturer's Operator's Manual (ref 1) should sufficiently prepare you with all the necessary skills to load and begin execution of simple programs. This section of the guide will discuss two methods of loading the program modules into the Model 100's BASIC Interpreter for execution. The first method involves downloading the programs from files stored in the student's personal account on the Naval Postgraduate School's IBM 3033 mainframe computer. The second method makes use of a standard cassette to load the program directly onto the Model 100 BASIC Interpreter. While downloading the program from the IBM mainframe is both slower and more complicated, it provides more flexibility in the distribution, accessibility, and maintenance of the program modules. In addition, the ability to exchange files with the mainframe is a valuable skill to learn and has applications in assisting the student in other areas of his/her education.

The Model 100 must be properly set up and prepared before any of the program modules can be loaded. The basic set up procedures are covered in the Operator's Manual (ref 1). One additional requirement is that due to the length of some of the modules, the Model 100's memory should be cleared prior to attempting either of the loading procedures. This is especially critical when downloading the programs from the IBM mainframe. When downloading via the Model 100's internal modem, the program modules are saved as ASCII files. These files must then be converted into BASIC format before execution can begin. While the Model 100's BASIC Interpreter does this conversion automatically, it does not dynamically erase the ASCII copy of the program. This results in more than doubling the amount of Random Access Memory required for each module (at least until the user manually erases the ASCII copy). Several of the modules are 14k bytes in length and must be broken down into two parts to load into the BASIC interpreter.

The requirement for a "clean" memory is not as strict when loading from a cassette recorder. The cassette copy can be loaded directly into the BASIC Interpreter, thereby eliminating the need for additional memory space and the need to divide any of the modules into segments for loading. In addition, use of the cassettes will allow for two or three modules to reside in the Model 100's memory simultaneously.

Downloading Executable Programs Using The Model 100's Internal Modem

In order to download any module, a copy must already reside in the student's virtual storage area on the IBM 3033. (Any student who does not already have a copy of each module on his/her "A" disk should consult the course instructor.) The actual downloading is accomplished using the Model 100's TELCOM program. TELCOM is one of five "instant on" programs which permanently reside in the read only memory of the Model 100. Before using the TELCOM program, it is necessary to provide the Model 100 access to a telephone line. The actual connection is made using the supplied modem connect cable and is discussed in detail in the Operator's Manual (ref 1, pg 76).

After connecting the Model 100 to a telephone line, turn the computer on and access the TELCOM program from the main menu display. You are now ready to begin the procedure to achieve a modem link with the IBM mainframe.

STEP 1: Access the TELCOM program and push function key F2. The word "CALL" will appear on the screen. Type the telephone access number of the IBM mainframe into the display after the word "CALL" as follows: 6463025<> (The less than (<) and the greater than (>) symbols following the number are required!)

STEP 2: Push the ENTER key and stand by as the computer automatically dials the number. When a connection is made, an audible beep will be heard and a message will appear on the screen regarding the status of the IBM mainframe. At this point, the user should check to ensure the Model 100 is set up to communicate in half-duplex. (This information is displayed on the bottom of the screen above "Position 4". Push function key F4 to cycle between half and full duplex as necessary.)

STEP 3: Push the ENTER key and a period (.) will appear at either of the screen margins. This period is the "ready" prompt. Log on to the mainframe following normal procedures. After log on is accomplished, you will be using the Model 100 as just another (teletype) terminal hooked into the IBM mainframe. The only significant differences from the terminals on campus are that you have a smaller set of CP/CMS commands available to use, and there is no full screen editing available. [Note: After initial logon, the user should proceed directly to CMS and make no attempt to display material in the "Virtual Reader" or to view the "News". The execution of these or similar functions may send a seemingly endless stream of information to the Model 100 which will significantly delay the module downloading process.]

STEP 4: Make the selection of which module you would like to download from the provided list. Push function key F2. The Model 100 will then query you for the title of the file you wish to create to accept the module to be downloaded. At this point, you can input any name up to six characters in length. It is suggested that the title be identical to the file name on the mainframe which is to be downloaded (i.e., MOD1, MOD3P1, etc.). Push ENTER after you type in the title to resume communications with the IBM mainframe.

STEP 5: Enter the command TYPE followed by the file name and file type of the module to be downloaded. (The file type of all the provided modules is "SCRIPT".)

Example: **Type MOD2 Script**

The selected program module will begin to scroll down the screen of the Model 100 as it is being loaded into an ASCII file with the title which was selected in Step 4.

STEP 6: The program has been completely entered into the Model 100's memory when the mainframe's "R" message appears on the screen and the scrolling stops. (This will take anywhere from three to nine minutes, depending on module length. Push function key F2 again to cease downloading. (Modules 3, 6, and 7 have been segmented into two parts and each module occupies two separate files on the IBM mainframe. To use these modules, both files must be downloaded to the Model 100. If you are attempting to download one of these segmented programs, return to Step 4 and repeat the procedures to download the second half of the program. Remember to use a different title for the name of the Model 100 file which will receive the second half of the program. Log off the IBM mainframe and exit TELCOM.

STEP 7: The Model 100's main menu should contain an ASCII file with the name specified in Step 4 (two files if it is a segmented program). Access this file and remove any extraneous data which may have been downloaded with the program module. This "garbage" is normally at the end of the file (although it is wise to check the top of the file also) and consists of the IBM's "R" message and any other messages which may have appeared on the screen while downloading. These extraneous remarks must be removed prior to loading the file into the Model 100's BASIC Interpreter.

STEP 8: Access the BASIC Interpreter from the main menu. Load the ASCII file with the command "LOAD FILENAME.DO".

Example: **LOAD "MOD2.DO"**

Push ENTER and the file will begin loading. After loading is complete, the screen will display the "OK" prompt. At this point, the user should erase the ASCII copy of the file to free additional memory. This is accomplished using the "KILL" command.

Example: **KILL "MOD2.DO"**

If the program has been segmented into two parts, load part one per the instructions above; "KILL" the ASCII copy of the first file, then load the second file with the command "MERGE RAM:FILENAME.DO".

Example:
```
LOAD "MOD3P1.DO"
KILL "MOD3P1.DO"
MERGE "RAM:MOD3P2.DO"
```

The program is now ready to be run. (If an error occurs during loading into the BASIC Interpreter, it is probably due to extraneous garbage which should have been removed in Step 7. Return to the main menu and repeat Steps 7 and 8.

Loading the Programs Via Cassette Recorder

Loading the program modules using a cassette recorder is much simpler and faster than using the modem. If you have access to a recorder, obtain a copy of the master tape which contains all eight program modules from the instructor. When using the cassette recorder to load, it is possible to load and save more than one module on the Model 100 at a time. This is due to the greatly reduced memory requirements for loading by reading the modules directly into the BASIC Interpreter from tape.

STEP 1: Using the instructions contained in the Operator's Manual (ref 1, pg 189), attach the cassette recorder to the Model 100.

STEP 2: Access the BASIC Interpreter from the main menu. Advance the cassette tape to the approximate starting position of the module to be loaded. Load the module directly into the BASIC Interpreter with the command CLOAD"FILENAME".

Example: **CLOAD"MOD2"**

Depress the PLAY button on the cassette recorder and the computer will search for the proper module file. When the file has been located, the screen will display the response FOUND"FILENAME".

Example: **FOUND"MOD2"**

The designated program will then be loaded into the BASIC Interpreter. When the cassette player ceases and the "OK" prompt appears on the screen, the program is ready to be run.

The actual loading time with this method ranges between one and two and one-half minutes, depending upon which module is being loaded. This is a great deal faster than the three to nine minutes required when downloading using the modem, and there is no additional time lost in the conversion from ASCII format to BASIC.

PART III: EXAMPLE PROGRAM RUNS

This section provides a printout of select screen displays used in solving an example problem from each of the topic areas covered by the program modules. All of the examples were chosen from Freund and Williams, Elementary Business Statistics: The Modern Approach; 4th Edition, unless otherwise noted. An understanding of certain key characteristics of the screen displays is necessary to effectively use these example listings.

1. Each screen has been tailored to fit the 40 character by 8 line display capability of the Model 100.
2. User input is displayed in italics in the listings following the question mark (?) prompt.
3. The user should be aware that only a portion of the innumerable screen displays encompassed by any one program module are listed. In fact, some of the more interesting screen displays which are pertinent to the example problems are not listed for various reasons (such as scrolling of data and temporary operator feedback messages). Comments have been inserted between the listings to alert the user to the absence of a particular screen display in the listings.
4. The final answer obtained through use of the program modules may differ from the results cited in

Freund and Williams. This is generally due to the increased accuracy of the computer.

5. In some instances, four zeros (0000) are input in response to a query to induce an intentional error in order to demonstrate the error checking facilities of the program modules.

6. Due to the variable length of certain key user input values, a few screen displays may experience "word chop" or the random division of complete words at the edge of the 40 character screen display.

Format of the Example Program Runs

A printout of the screen displays follows the statement of select example problems. Each topic area within a module is sequentially exercised and listed. In some cases, it may appear that the same screen display has been printed twice. A closer examination will reveal additional data in the latter screen. This apparent duplication is manifest only in the listings and represents an attempt to convey in hardcopy the stepwise sequential queries used to interact with the user. Finally, the user should be aware that these listings are provided only to give a taste of each module. The only effective way to gain a full appreciation of each module is through actual use. In many ways, it is much easier to jump right in and use the modules than to try and "understand" the listings. Remember, mastery of the subject material, not the programs, is the ultimate objective.

EXAMPLE PROGRAM RUN INDEX

Module 1

Permutations....................................131

Combinations....................................133

Bayes' Theorem..................................134

Module 2 (Generation of Random Samples)

Finite Populations..............................138

Exponential.....................................141

Normal..143

Uniform...145

Module 3 (Applications of Various Distributions

Binomial Distributions..........................148

Geometric Distributions.........................151

Poisson Distributions...........................153

Exponential Distributions.......................156

Module 4

Central Limit Theorem...........................159

Applications of the Normal Distribution.........162

Module 5

Maximum Error of an Estimate....................166

Determination of Minimum Sample Size to
Achieve a Given Level of Error..................168

Confidence Interval About the Mean..............171

Hypothesis Testing of the Mean..................175

Module 6

Confidence Interval About the Standard Deviation....................................180

Hypothesis Testing of the Standard Deviation...183

Module 7 (Linear Regression and Analysis)

Hypothesis Testing of Alpha and Beta...........190

Confidence Interval for Alpha or Beta..........193

Confidence Interval for the Mean of Y at X Naught..................................195

Limits of Predictions..........................197

Estimates of Y When Given X....................199

Module 8 (Multiple Regression)

Multiple Regression............................201

Significance Testing for R.....................203

Coefficient of Determination...................207

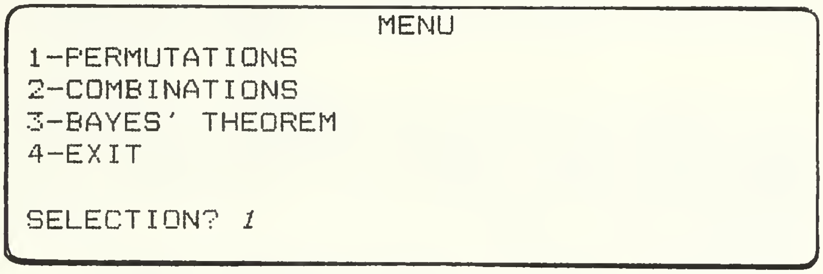
Estimates of Y When Given X1, X2...............209

MODULE 1

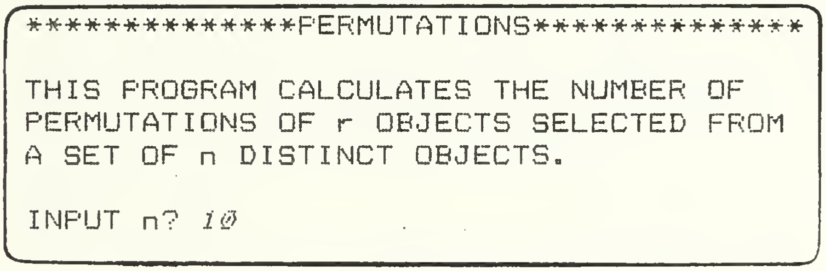
EXAMPLES AND SCREEN DISPLAYS

PERMUTATIONS EXAMPLE: Find the number of ways in which three of ten real estate salespersons can be ranked first, second, and third according to market knowledgeability.

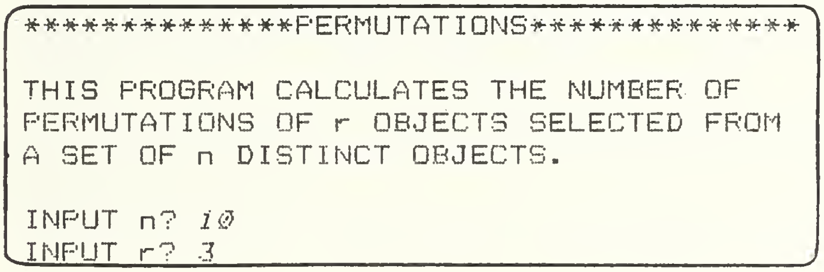
SOLUTION: For n=10 and r=3 the program yields (Ref 2, pg 108)



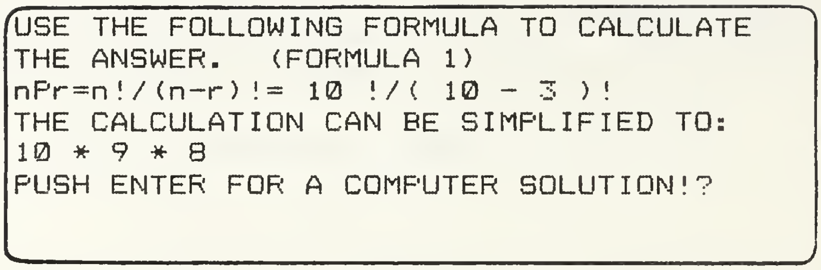
```
                    MENU
1-PERMUTATIONS
2-COMBINATIONS
3-BAYES' THEOREM
4-EXIT

SELECTION? 1
```



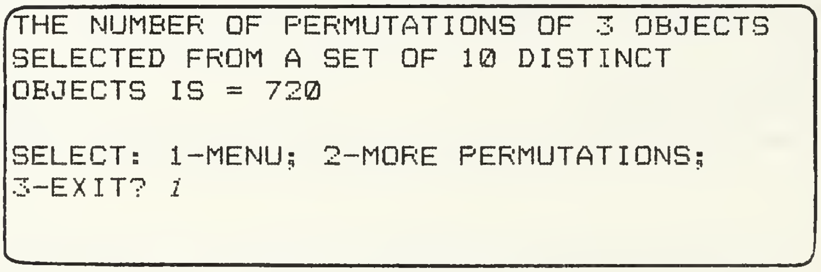
```
**************PERMUTATIONS**************

THIS PROGRAM CALCULATES THE NUMBER OF
PERMUTATIONS OF r OBJECTS SELECTED FROM
A SET OF n DISTINCT OBJECTS.

INPUT n? 10
```



```
**************PERMUTATIONS**************

THIS PROGRAM CALCULATES THE NUMBER OF
PERMUTATIONS OF r OBJECTS SELECTED FROM
A SET OF n DISTINCT OBJECTS.

INPUT n? 10
INPUT r? 3
```



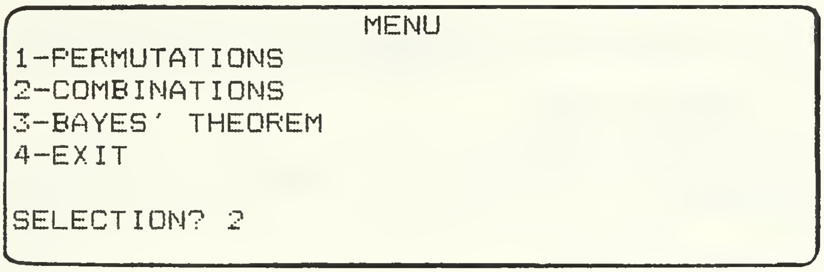
```
USE THE FOLLOWING FORMULA TO CALCULATE
THE ANSWER.  (FORMULA 1)
nPr=n!/(n-r)!= 10 !/( 10 - 3 )!
THE CALCULATION CAN BE SIMPLIFIED TO:
10 * 9 * 8
PUSH ENTER FOR A COMPUTER SOLUTION!?
```



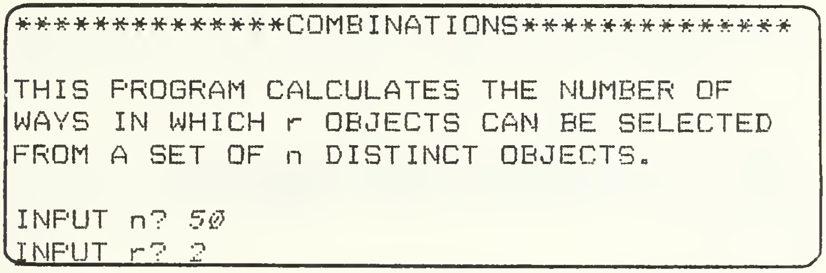
```
THE NUMBER OF PERMUTATIONS OF 3 OBJECTS
SELECTED FROM A SET OF 10 DISTINCT
OBJECTS IS = 720

SELECT: 1-MENU; 2-MORE PERMUTATIONS;
3-EXIT? 1
```

COMBINATIONS EXAMPLE: In how many ways can a dean choose 2 of 50 faculty members to review a student grade appeal?

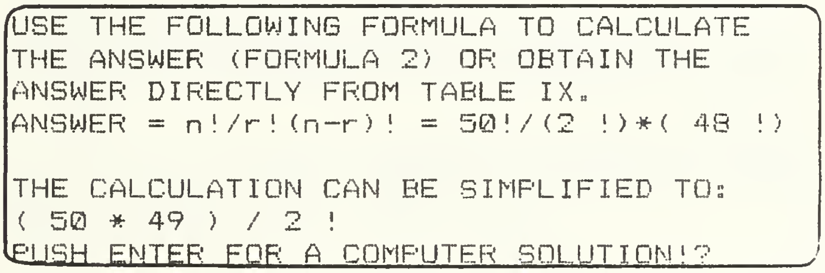
SOLUTION: For n=50 and r=2, the first program yields (Ref 2, pg 110)



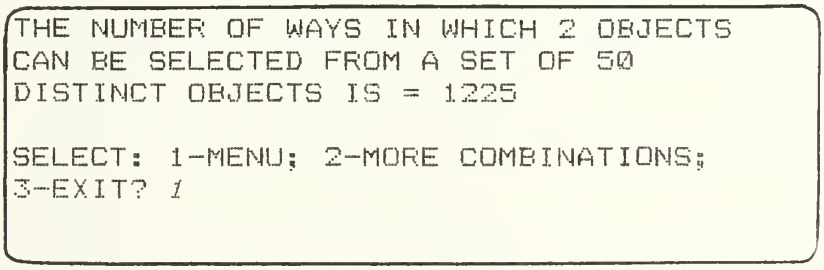
```
                    MENU
1-PERMUTATIONS
2-COMBINATIONS
3-BAYES' THEOREM
4-EXIT

SELECTION? 2
```



```
**************COMBINATIONS**************

THIS PROGRAM CALCULATES THE NUMBER OF
WAYS IN WHICH r OBJECTS CAN BE SELECTED
FROM A SET OF n DISTINCT OBJECTS.

INPUT n? 50
INPUT r? 2
```

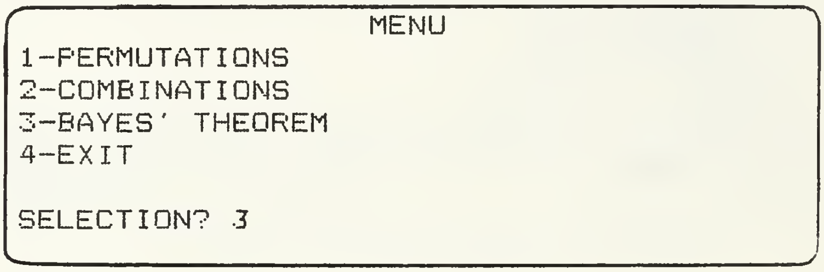


```
USE THE FOLLOWING FORMULA TO CALCULATE
THE ANSWER (FORMULA 2) OR OBTAIN THE
ANSWER DIRECTLY FROM TABLE IX.
ANSWER = n!/r!(n-r)! = 50!/(2 !)*( 48 !)

THE CALCULATION CAN BE SIMPLIFIED TO:
( 50 * 49 ) / 2 !
PUSH ENTER FOR A COMPUTER SOLUTION!?
```

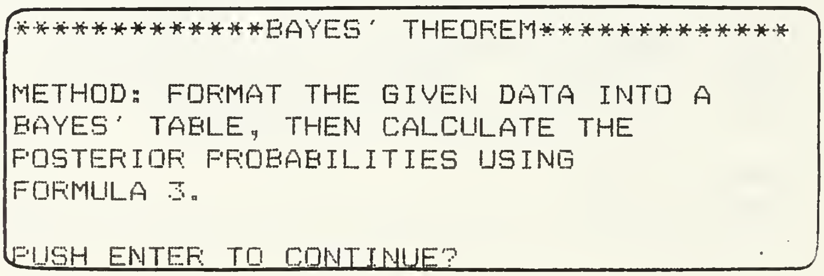


```
THE NUMBER OF WAYS IN WHICH 2 OBJECTS
CAN BE SELECTED FROM A SET OF 50
DISTINCT OBJECTS IS = 1225

SELECT: 1-MENU; 2-MORE COMBINATIONS;
3-EXIT? 1
```

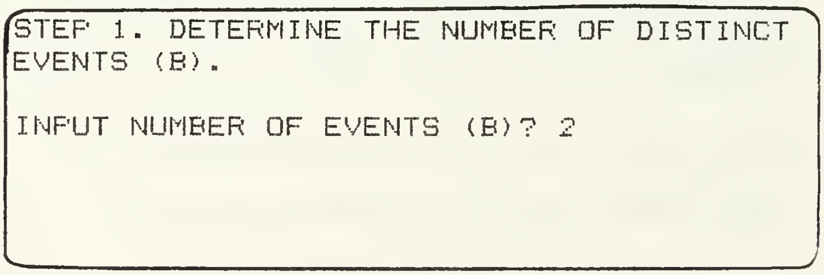
BAYES' THEOREM EXAMPLE: The probability that a one-car accident is due to faulty brakes is 0.04, the probability that a one-car accident is correctly attributed to faulty brakes is 0.82, and the probability that a one-car accident is incorrectly attributed to faulty brakes is 0.03. What is the probability that a one-car accident attributed to faulty brakes was actually due to faulty brakes? (Ref 2, pg 241)



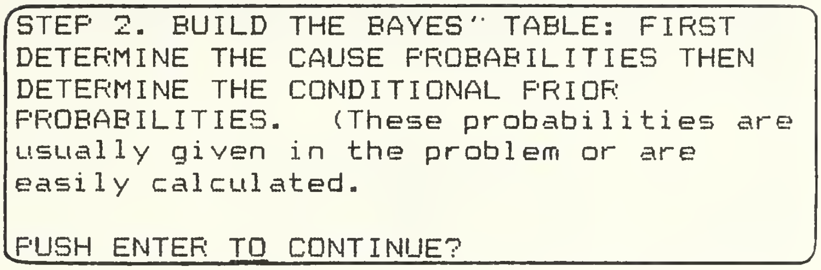
```
                         MENU
1-PERMUTATIONS
2-COMBINATIONS
3-BAYES' THEOREM
4-EXIT

SELECTION? 3
```



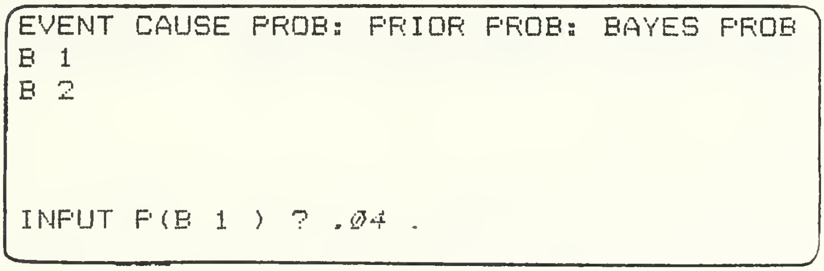
```
*************BAYES' THEOREM*************

METHOD: FORMAT THE GIVEN DATA INTO A
BAYES' TABLE, THEN CALCULATE THE
POSTERIOR PROBABILITIES USING
FORMULA 3.

PUSH ENTER TO CONTINUE?
```



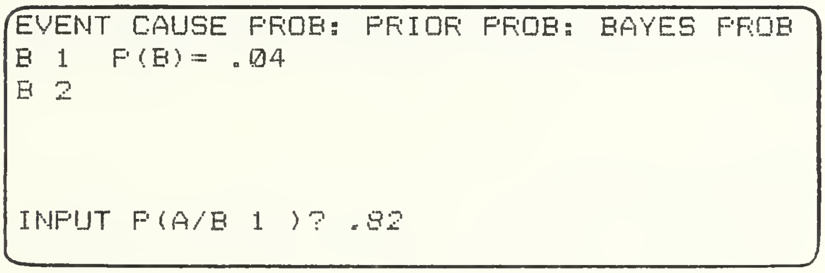
```
STEP 1. DETERMINE THE NUMBER OF DISTINCT
EVENTS (B).

INPUT NUMBER OF EVENTS (B)? 2
```



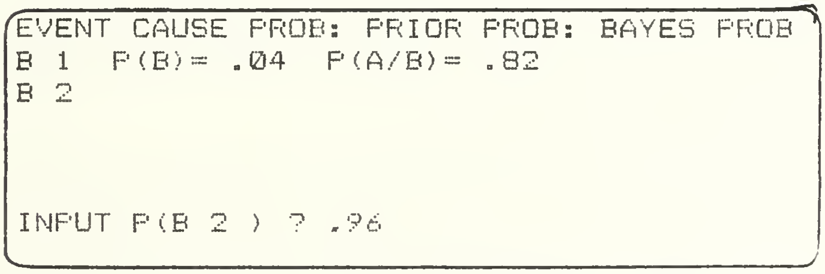
```
STEP 2. BUILD THE BAYES'' TABLE: FIRST
DETERMINE THE CAUSE PROBABILITIES THEN
DETERMINE THE CONDITIONAL PRIOR
PROBABILITIES.  (These probabilities are
usually given in the problem or are
easily calculated.

PUSH ENTER TO CONTINUE?
```



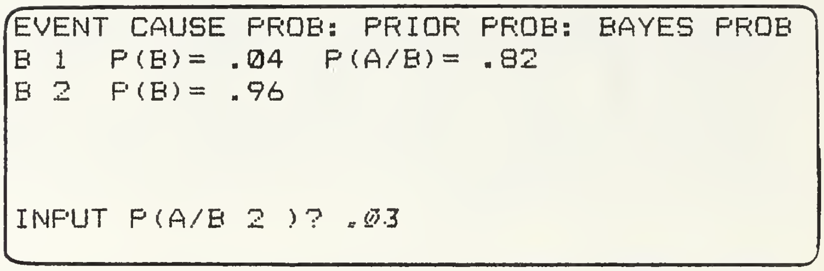
```
EVENT CAUSE PROB: PRIOR PROB: BAYES PROB
B 1
B 2




INPUT P(B 1 ) ? .04 .
```



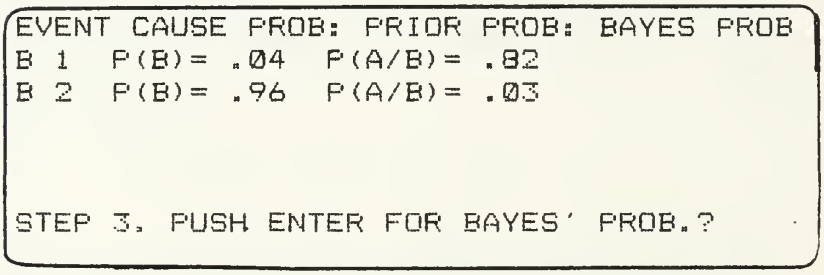
```
EVENT CAUSE PROB: PRIOR PROB: BAYES PROB
B 1  P(B)= .04
B 2




INPUT P(A/B 1 )? .82
```



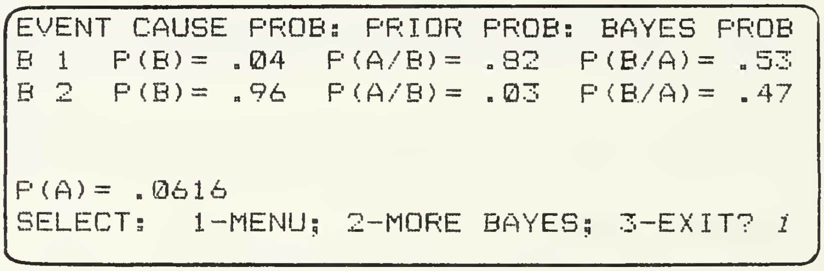
```
EVENT CAUSE PROB: PRIOR PROB: BAYES PROB
B 1  P(B)= .04   P(A/B)= .82
B 2




INPUT P(B 2 ) ? .96
```



```
EVENT CAUSE PROB: PRIOR PROB: BAYES PROB
B 1   P(B)= .04   P(A/B)= .82
B 2   P(B)= .96




INPUT P(A/B 2 )? .03
```



```
EVENT CAUSE PROB: PRIOR PROB: BAYES PROB
B 1   P(B)= .04   P(A/B)= .82
B 2   P(B)= .96   P(A/B)= .03




STEP 3. PUSH ENTER FOR BAYES' PROB.?
```

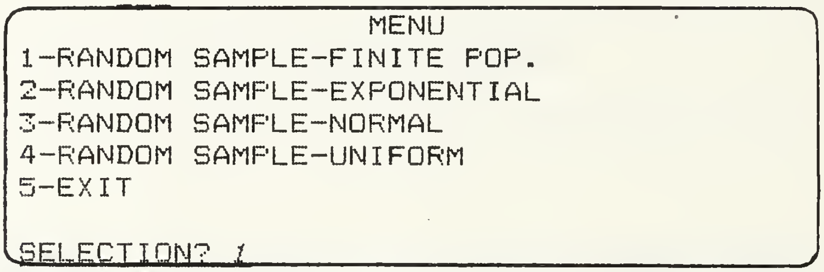


```
EVENT CAUSE PROB: PRIOR PROB: BAYES PROB
B 1   P(B)= .04   P(A/B)= .82  P(B/A)= .53
B 2   P(B)= .96   P(A/B)= .03  P(B/A)= .47


P(A)= .0616
SELECT:  1-MENU; 2-MORE BAYES; 3-EXIT? 1
```

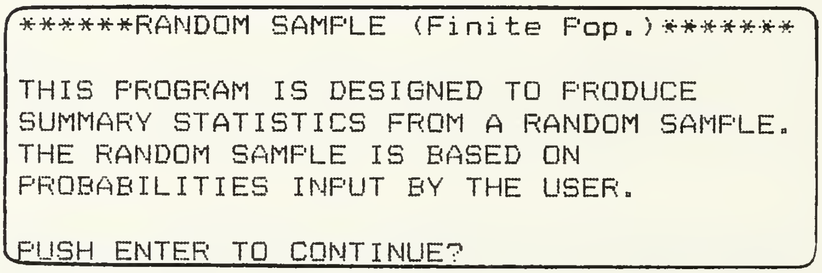
MODULE 2

EXAMPLES AND SCREEN DISPLAYS

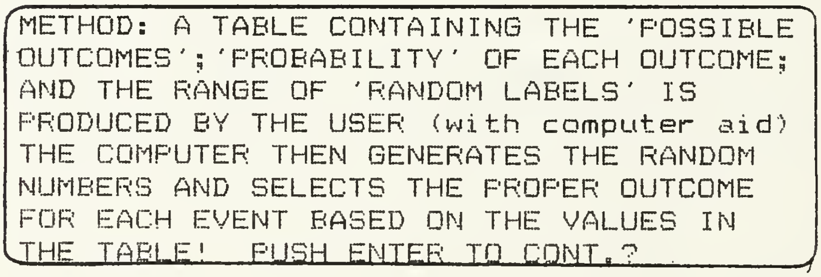
FINITE POPULATIONS EXAMPLE: Simulate with random numbers the number of heads expected in 20 tosses of three fair coins. (Ref 2, pg 295)



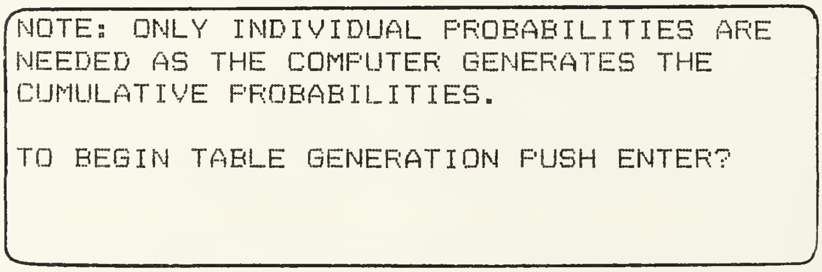
```
                    MENU
1-RANDOM SAMPLE-FINITE POP.
2-RANDOM SAMPLE-EXPONENTIAL
3-RANDOM SAMPLE-NORMAL
4-RANDOM SAMPLE-UNIFORM
5-EXIT

SELECTION? 1
```



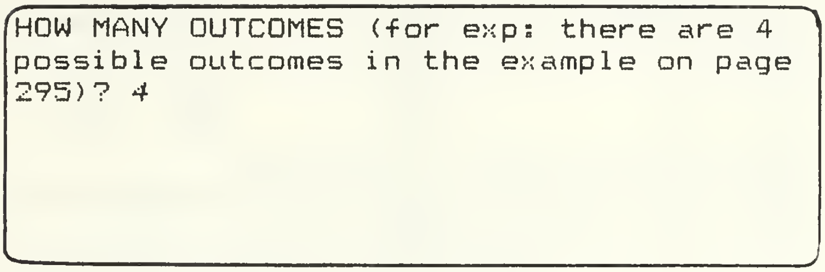
```
******RANDOM SAMPLE (Finite Pop.)*******

THIS PROGRAM IS DESIGNED TO PRODUCE
SUMMARY STATISTICS FROM A RANDOM SAMPLE.
THE RANDOM SAMPLE IS BASED ON
PROBABILITIES INPUT BY THE USER.

PUSH ENTER TO CONTINUE?
```



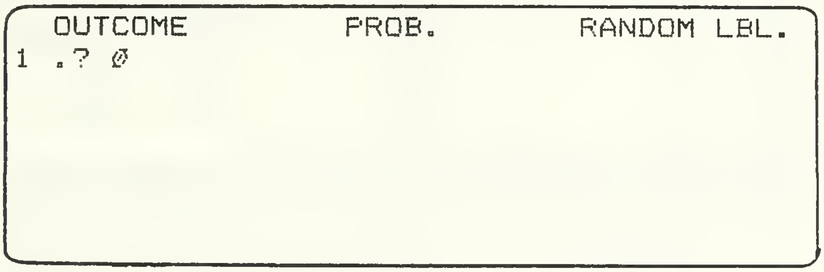
```
METHOD: A TABLE CONTAINING THE 'POSSIBLE
OUTCOMES';'PROBABILITY' OF EACH OUTCOME;
AND THE RANGE OF 'RANDOM LABELS' IS
PRODUCED BY THE USER (with computer aid)
THE COMPUTER THEN GENERATES THE RANDOM
NUMBERS AND SELECTS THE PROPER OUTCOME
FOR EACH EVENT BASED ON THE VALUES IN
THE TABLE!  PUSH ENTER TO CONT.?
```



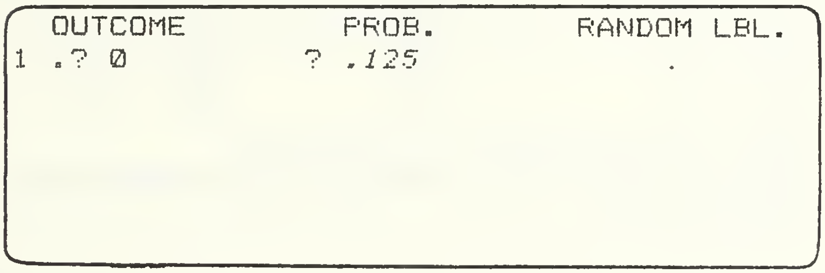
```
NOTE: ONLY INDIVIDUAL PROBABILITIES ARE
NEEDED AS THE COMPUTER GENERATES THE
CUMULATIVE PROBABILITIES.

TO BEGIN TABLE GENERATION PUSH ENTER?
```



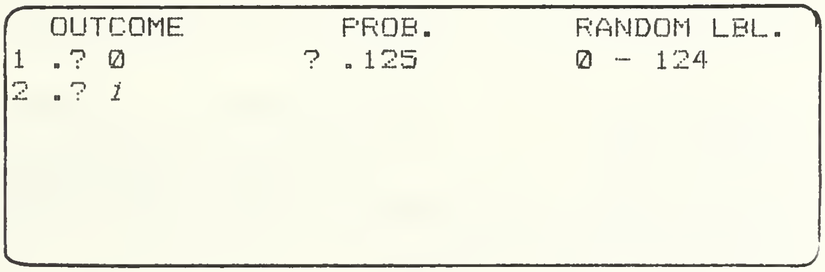
```
HOW MANY OUTCOMES (for exp: there are 4
possible outcomes in the example on page
295)? 4
```



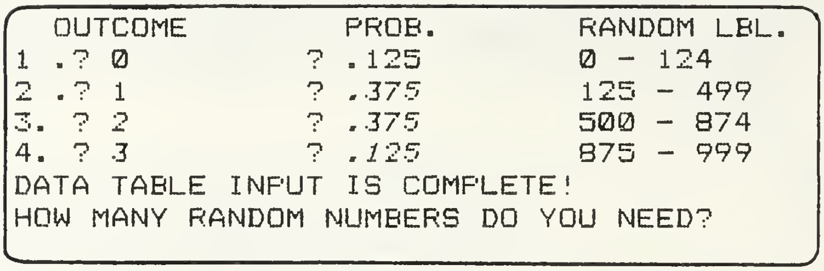
```
   OUTCOME            PROB.          RANDOM LBL.
1 .? 0
```



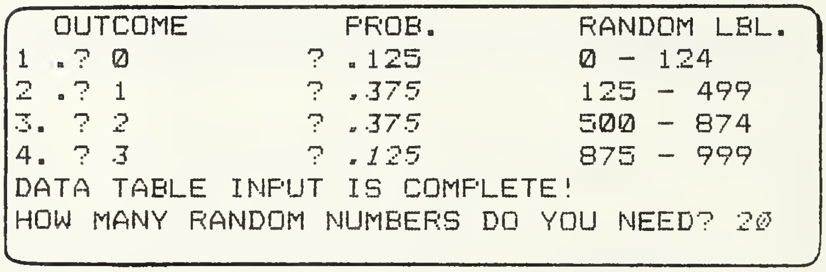
```
   OUTCOME            PROB.          RANDOM LBL.
1 .? 0             ? .125                 .
```



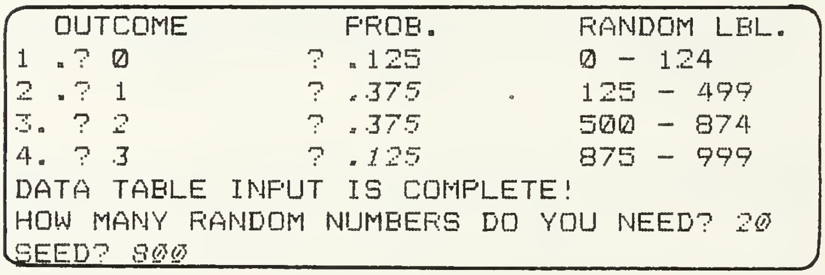
```
   OUTCOME            PROB.          RANDOM LBL.
1 .? 0             ? .125        0 - 124
2 .? 1
```



```
  OUTCOME            PROB.           RANDOM LBL.
1 .? 0             ? .125            0 - 124
2 .? 1             ? .375            125 - 499
3. ? 2             ? .375            500 - 874
4. ? 3             ? .125            875 - 999
DATA TABLE INPUT IS COMPLETE!
HOW MANY RANDOM NUMBERS DO YOU NEED?
```

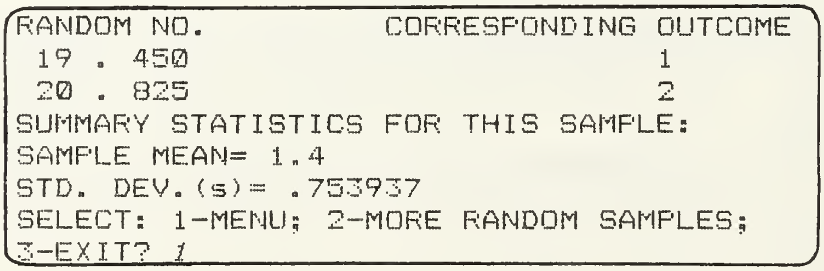


```
  OUTCOME            PROB.           RANDOM LBL.
1 .? 0             ? .125            0 - 124
2 .? 1             ? .375            125 - 499
3. ? 2             ? .375            500 - 874
4. ? 3             ? .125            875 - 999
DATA TABLE INPUT IS COMPLETE!
HOW MANY RANDOM NUMBERS DO YOU NEED? 20
```



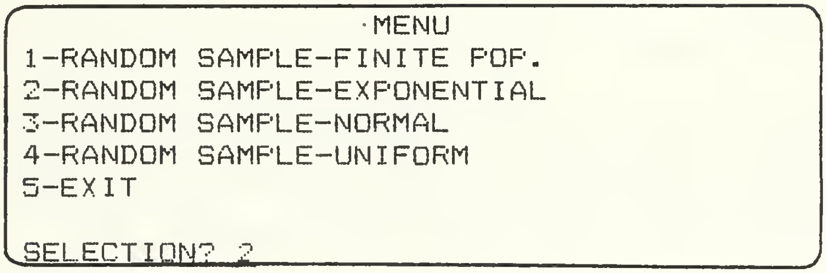
```
  OUTCOME            PROB.           RANDOM LBL.
1 .? 0             ? .125            0 - 124
2 .? 1             ? .375            125 - 499
3. ? 2             ? .375            500 - 874
4. ? 3             ? .125            875 - 999
DATA TABLE INPUT IS COMPLETE!
HOW MANY RANDOM NUMBERS DO YOU NEED? 20
SEED? 800
```

At this point, the program generates 20 random numbers which are scrolled down the screen along with the calculated corresponding outcome.

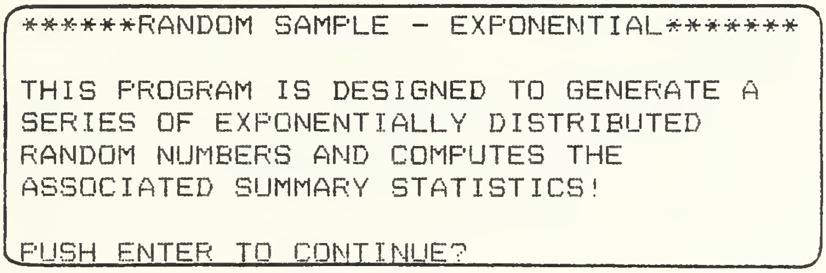


```
RANDOM NO.            CORRESPONDING OUTCOME
 19 . 450                             1
 20 . 825                             2
SUMMARY STATISTICS FOR THIS SAMPLE:
SAMPLE MEAN= 1.4
STD. DEV.(s)= .753937
SELECT: 1-MENU; 2-MORE RANDOM SAMPLES;
3-EXIT? 1
```

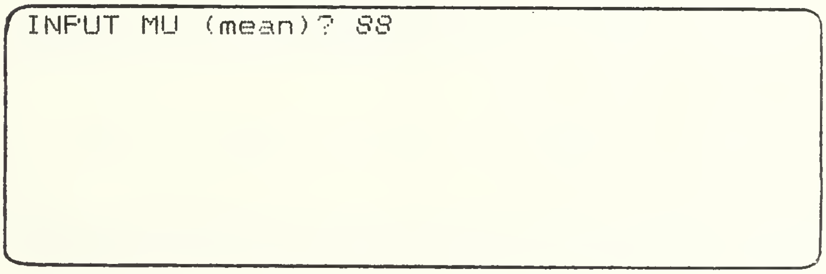
EXPONENTIAL EXAMPLE: Generate 20 exponentially distributed random numbers with a mean of 88. Use a seed of 75. Calculate the mean and standard deviation of the random sample.



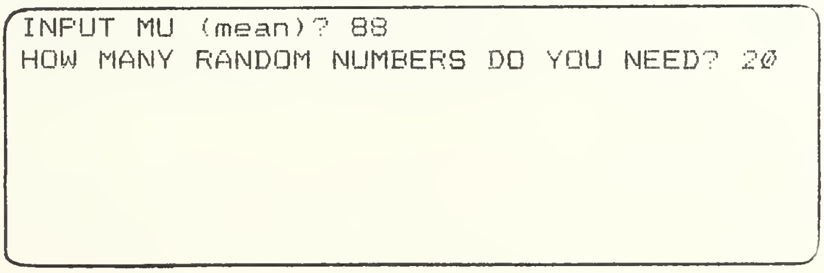
```
                  MENU
1-RANDOM SAMPLE-FINITE POP.
2-RANDOM SAMPLE-EXPONENTIAL
3-RANDOM SAMPLE-NORMAL
4-RANDOM SAMPLE-UNIFORM
5-EXIT

SELECTION? 2
```



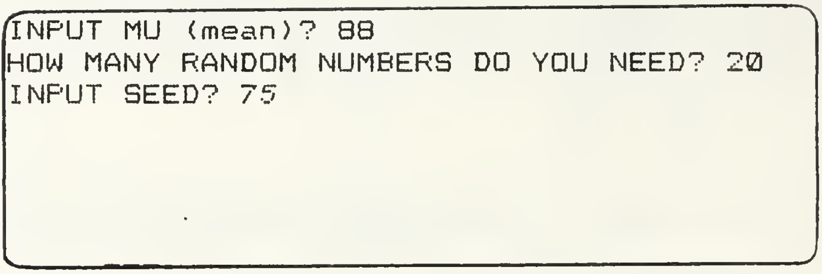
```
******RANDOM SAMPLE - EXPONENTIAL*******

THIS PROGRAM IS DESIGNED TO GENERATE A
SERIES OF EXPONENTIALLY DISTRIBUTED
RANDOM NUMBERS AND COMPUTES THE
ASSOCIATED SUMMARY STATISTICS!

PUSH ENTER TO CONTINUE?
```



```
INPUT MU (mean)? 88
```

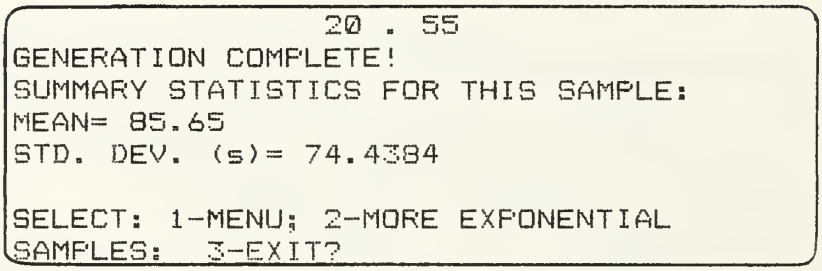


```
INPUT MU (mean)? 88
HOW MANY RANDOM NUMBERS DO YOU NEED? 20
```



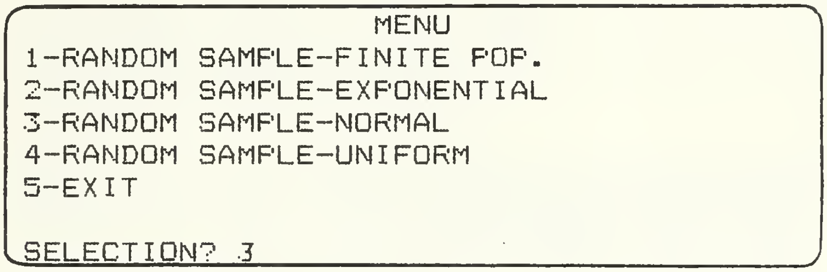
```
INPUT MU (mean)? 88
HOW MANY RANDOM NUMBERS DO YOU NEED? 20
INPUT SEED? 75
```

At this point, the program generates 20 random numbers which are developed around the input parameters. These numbers are then scrolled down the screen. The user can stop the scrolling action at any point with the **PAUSE** key.

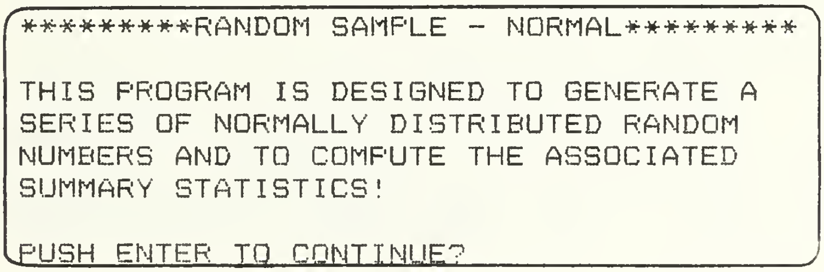


```
                    20 . 55
GENERATION COMPLETE!
SUMMARY STATISTICS FOR THIS SAMPLE:
MEAN= 85.65
STD. DEV. (s)= 74.4384

SELECT: 1-MENU; 2-MORE EXPONENTIAL
SAMPLES:  3-EXIT?
```

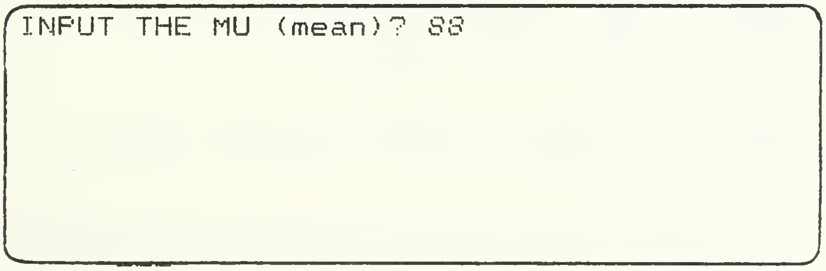
NORMAL EXAMPLE: Generate 20 normally distributed random numbers from a population with a mean of 88 and a standard deviation of 5. Use 75 as a seed. Display the mean and the standard deviation for the sample.



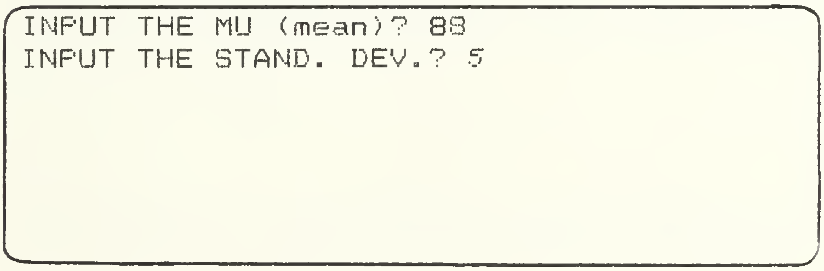
```
                    MENU
1-RANDOM SAMPLE-FINITE POP.
2-RANDOM SAMPLE-EXPONENTIAL
3-RANDOM SAMPLE-NORMAL
4-RANDOM SAMPLE-UNIFORM
5-EXIT

SELECTION? 3
```



```
*********RANDOM SAMPLE - NORMAL*********

THIS PROGRAM IS DESIGNED TO GENERATE A
SERIES OF NORMALLY DISTRIBUTED RANDOM
NUMBERS AND TO COMPUTE THE ASSOCIATED
SUMMARY STATISTICS!

PUSH ENTER TO CONTINUE?
```



```
INPUT THE MU (mean)? 88
```



```
INPUT THE MU (mean)? 88
INPUT THE STAND. DEV.? 5
INPUT THE MU (mean)? 88
INPUT THE STAND. DEV.? 5
HOW MANY RANDOM SAMPLES DO YOU NEED? 20
```

```
INPUT THE MU (mean)? 88
INPUT THE STAND. DEV.? 5
HOW MANY RANDOM SAMPLES DO YOU NEED? 20
INPUT SEED (for random sample)? 75
```

At this point, the program generates and displays 20 exponentially distributed random numbers using the input parameters. The random numbers are scrolled down the screen for user perusal. (Use the PAUSE key to halt the scrolling action.)

```
SUMMARY STATISTICS FOR THIS SAMPLE:
SAMPLE MEAN (Xbar)= 89.514
SAMPLE STDV.(s)= 5.81231

SELECT: 1-MENU; 2-MORE NORMALLY DIST.
SAMPLES; 3-EXIT? 1
```

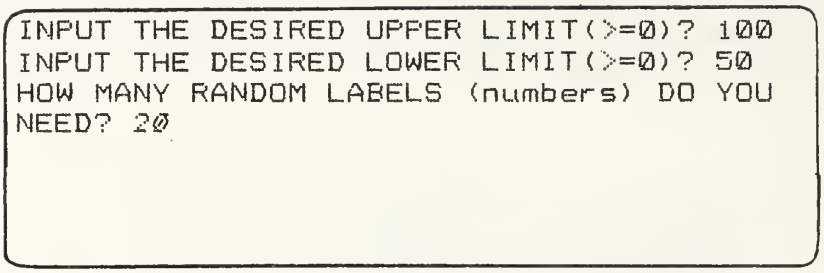
UNIFORM EXAMPLE: Generate 20 uniformly distributed random numbers between 50 and 100. Start with a seed of 75. Find the mean and the standard deviation of the generated sample.

```
                    MENU
1-RANDOM SAMPLE-FINITE POP.
2-RANDOM SAMPLE-EXPONENTIAL
3-RANDOM SAMPLE-NORMAL
4-RANDOM SAMPLE-UNIFORM
5-EXIT

SELECTION? 4
```

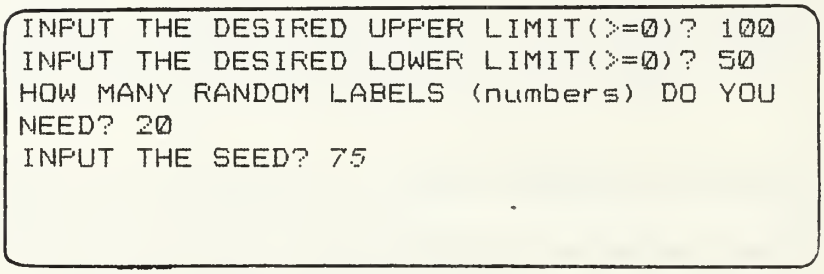
```
********RANDOM SAMPLE - UNIFORM*********

THIS PROGRAM IS DESIGNED TO GENERATE A
SERIES OF UNIFORMLY DISTRIBUTED RANDOM
NUMBERS AND TO COMPUTE THE ASSOCIATED
SUMMARY STATISTICS!

PUSH ENTER TO CONTINUE?
```

```
INPUT THE DESIRED UPPER LIMIT(>=0)? 100
```

```
INPUT THE DESIRED UPPER LIMIT(>=0)? 100
INPUT THE DESIRED LOWER LIMIT(>=0)? 50
```

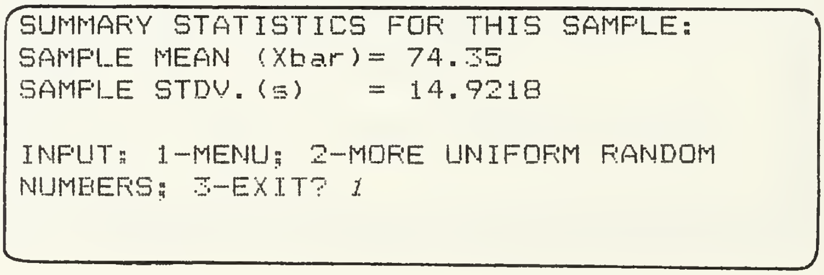


```
INPUT THE DESIRED UPPER LIMIT(>=0)? 100
INPUT THE DESIRED LOWER LIMIT(>=0)? 50
HOW MANY RANDOM LABELS (numbers) DO YOU
NEED? 20
```



```
INPUT THE DESIRED UPPER LIMIT(>=0)? 100
INPUT THE DESIRED LOWER LIMIT(>=0)? 50
HOW MANY RANDOM LABELS (numbers) DO YOU
NEED? 20
INPUT THE SEED? 75
```

At this point, the program generates and displays 20 uniformly distributed random numbers using the input parameters. The random numbers are scrolled down the screen for user perusal. (Use the **PAUSE** key to halt the scrolling action.)



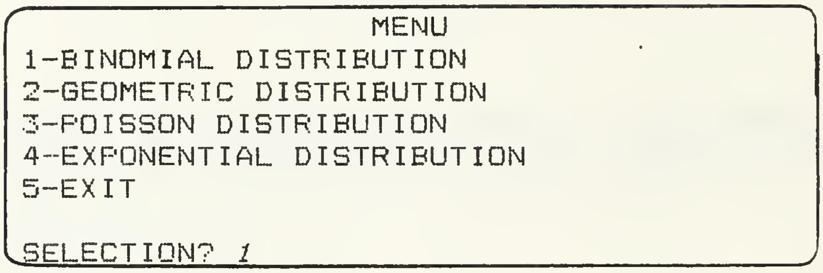
```
SUMMARY STATISTICS FOR THIS SAMPLE:
SAMPLE MEAN (Xbar)= 74.35
SAMPLE STDV.(s)    = 14.9218

INPUT: 1-MENU; 2-MORE UNIFORM RANDOM
NUMBERS; 3-EXIT? 1
```

MODULE 3

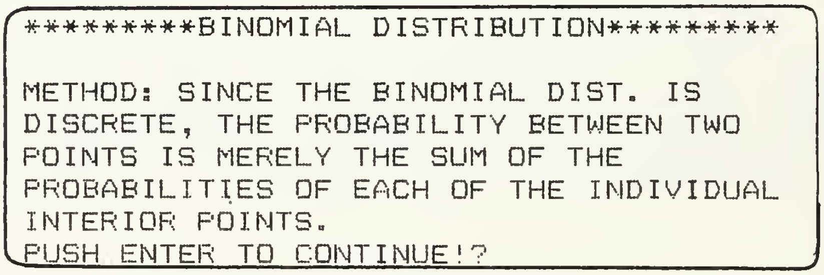
EXAMPLES AND SCREEN DISPLAYS

BINOMIAL DISTRIBUTION EXAMPLE: If the probability is 0.20 that any one shoplifter will get caught, what is the probability that in a random sample of eight shoplifters, three will get caught?

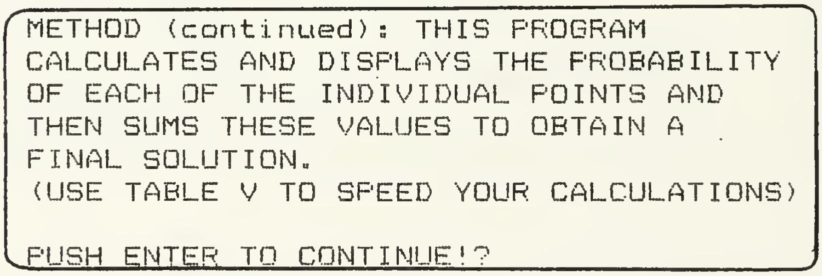
SOLUTION: Substitute $x=3$, $n=8$, $p=0.20$ into the program. (Ref 2, pg 214)



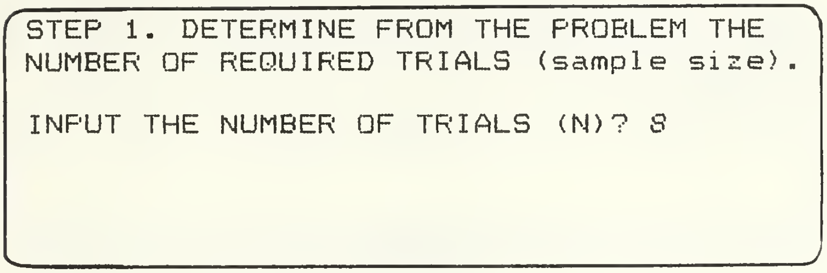
```
                   MENU
1-BINOMIAL DISTRIBUTION
2-GEOMETRIC DISTRIBUTION
3-POISSON DISTRIBUTION
4-EXPONENTIAL DISTRIBUTION
5-EXIT

SELECTION? 1
```



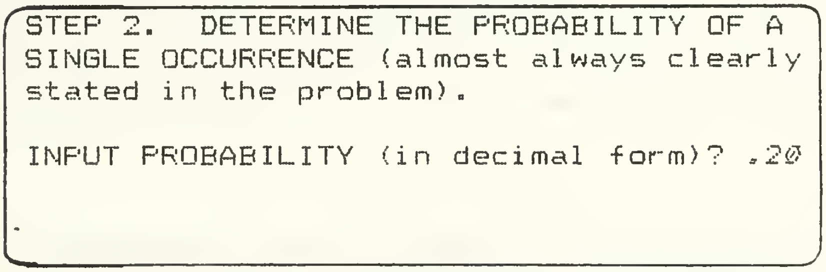
```
*********BINOMIAL DISTRIBUTION*********

METHOD: SINCE THE BINOMIAL DIST. IS
DISCRETE, THE PROBABILITY BETWEEN TWO
POINTS IS MERELY THE SUM OF THE
PROBABILITIES OF EACH OF THE INDIVIDUAL
INTERIOR POINTS.
PUSH ENTER TO CONTINUE!?
```



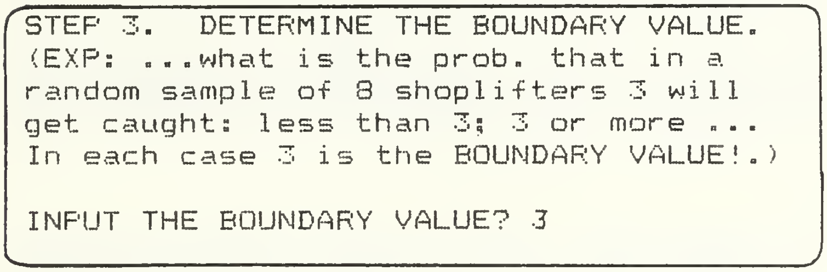
```
METHOD (continued): THIS PROGRAM
CALCULATES AND DISPLAYS THE PROBABILITY
OF EACH OF THE INDIVIDUAL POINTS AND
THEN SUMS THESE VALUES TO OBTAIN A
FINAL SOLUTION.
(USE TABLE V TO SPEED YOUR CALCULATIONS)

PUSH ENTER TO CONTINUE!?
```



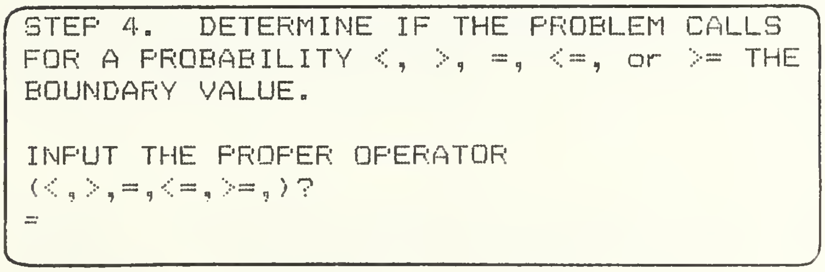
```
STEP 1. DETERMINE FROM THE PROBLEM THE
NUMBER OF REQUIRED TRIALS (sample size).

INPUT THE NUMBER OF TRIALS (N)? 8
```



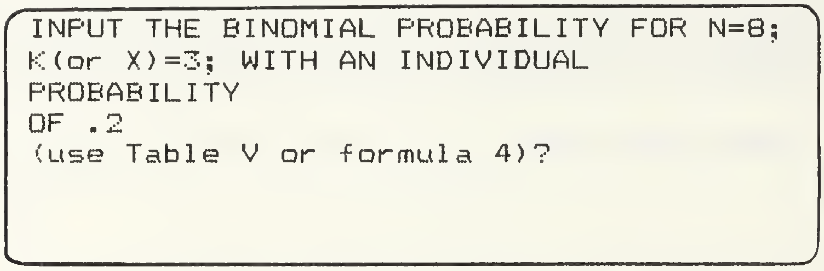
```
STEP 2.  DETERMINE THE PROBABILITY OF A
SINGLE OCCURRENCE (almost always clearly
stated in the problem).

INPUT PROBABILITY (in decimal form)? .20
```



```
STEP 3.  DETERMINE THE BOUNDARY VALUE.
(EXP: ...what is the prob. that in a
random sample of 8 shoplifters 3 will
get caught: less than 3; 3 or more ...
In each case 3 is the BOUNDARY VALUE!.)

INPUT THE BOUNDARY VALUE? 3
```

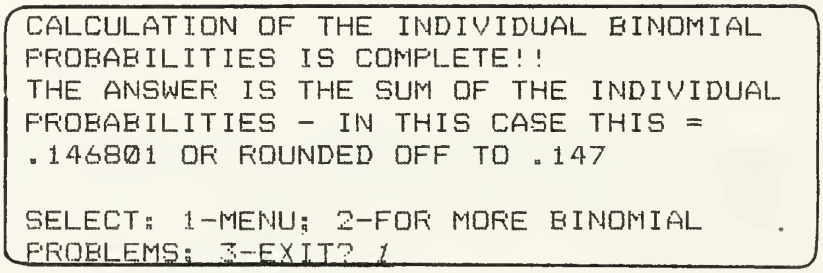


```
STEP 4.  DETERMINE IF THE PROBLEM CALLS
FOR A PROBABILITY <, >, =, <=, or >= THE
BOUNDARY VALUE.

INPUT THE PROPER OPERATOR
(<,>,=,<=,>=,)?
=
```



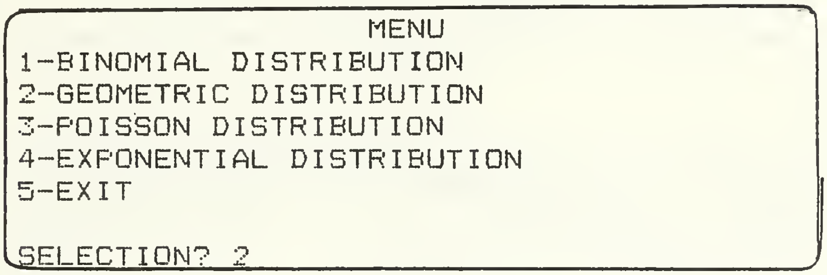
```
INPUT THE BINOMIAL PROBABILITY FOR N=8;
K(or X)=3; WITH AN INDIVIDUAL
PROBABILITY
OF .2
(use Table V or formula 4)?
```

The program will check the user input for this value and provide feedback and error corrections as necessary. In addition, data pertinent to the calculations is scrolled down the screen and each individual binominal probability is calculated by the user and the program. This continues until all the necessary binominal calculations have been completed.

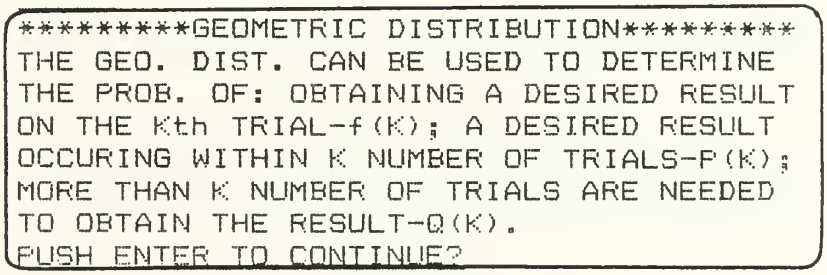


```
CALCULATION OF THE INDIVIDUAL BINOMIAL
PROBABILITIES IS COMPLETE!!
THE ANSWER IS THE SUM OF THE INDIVIDUAL
PROBABILITIES - IN THIS CASE THIS =
.146801 OR ROUNDED OFF TO .147

SELECT: 1-MENU; 2-FOR MORE BINOMIAL
PROBLEMS; 3-EXIT? 1
```

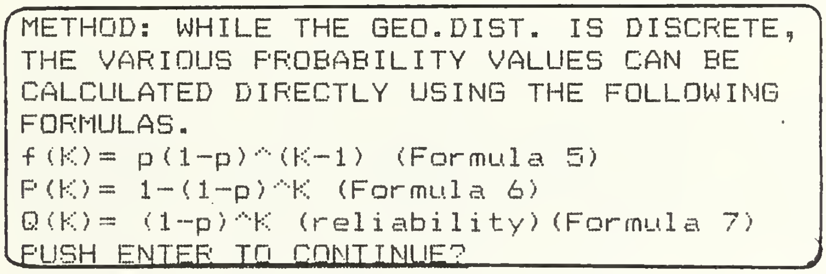
GEOMETRIC EXAMPLE: When taping a television commercial, the probability that a certain actor will get his lines straight on any one take is 0.40. What is the probability that this actor will get his lines straight for the first time on the fourth take? (Ref 2, pg 218)



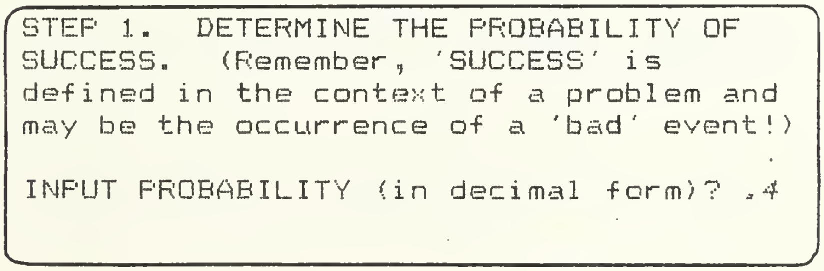
```
                    MENU
1-BINOMIAL DISTRIBUTION
2-GEOMETRIC DISTRIBUTION
3-POISSON DISTRIBUTION
4-EXPONENTIAL DISTRIBUTION
5-EXIT

SELECTION? 2
```



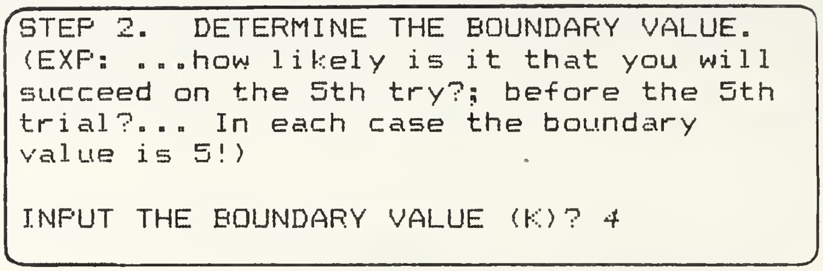
```
*********GEOMETRIC DISTRIBUTION*********
THE GEO. DIST. CAN BE USED TO DETERMINE
THE PROB. OF: OBTAINING A DESIRED RESULT
ON THE Kth TRIAL-f(K); A DESIRED RESULT
OCCURING WITHIN K NUMBER OF TRIALS-P(K);
MORE THAN K NUMBER OF TRIALS ARE NEEDED
TO OBTAIN THE RESULT-Q(K).
PUSH ENTER TO CONTINUE?
```



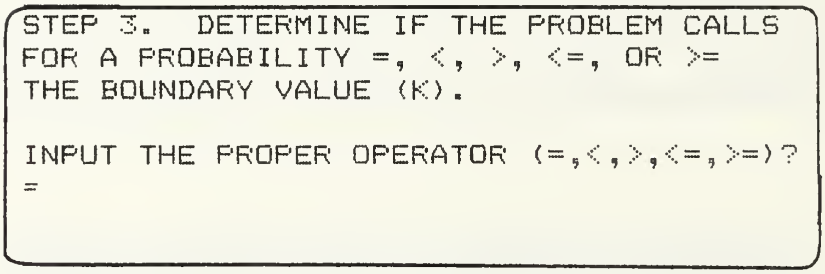
```
METHOD: WHILE THE GEO.DIST. IS DISCRETE,
THE VARIOUS PROBABILITY VALUES CAN BE
CALCULATED DIRECTLY USING THE FOLLOWING
FORMULAS.
f(K)= p(1-p)^(K-1)  (Formula 5)
P(K)= 1-(1-p)^K  (Formula 6)
Q(K)= (1-p)^K  (reliability)(Formula 7)
PUSH ENTER TO CONTINUE?
```



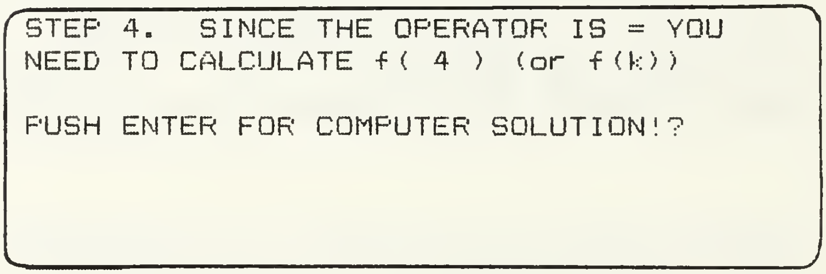
```
STEP 1.  DETERMINE THE PROBABILITY OF
SUCCESS.  (Remember, 'SUCCESS' is
defined in the context of a problem and
may be the occurrence of a 'bad' event!)

INPUT PROBABILITY (in decimal form)? .4
```



```
STEP 2.  DETERMINE THE BOUNDARY VALUE.
(EXP: ...how likely is it that you will
succeed on the 5th try?; before the 5th
trial?... In each case the boundary
value is 5!)

INPUT THE BOUNDARY VALUE (K)? 4
```

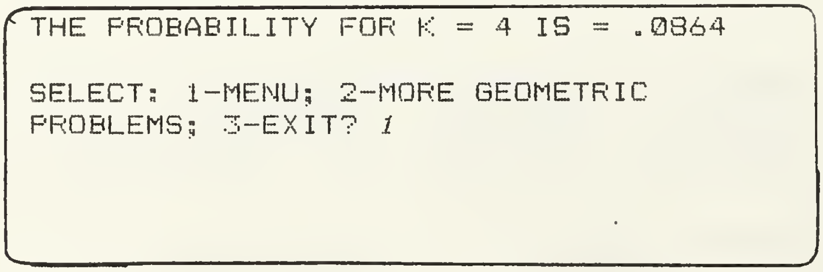


```
STEP 3.  DETERMINE IF THE PROBLEM CALLS
FOR A PROBABILITY =, <, >, <=, OR >=
THE BOUNDARY VALUE (K).

INPUT THE PROPER OPERATOR (=,<,>,<=,>=)?
=
```



```
STEP 4.  SINCE THE OPERATOR IS = YOU
NEED TO CALCULATE f( 4 ) (or f(k))

PUSH ENTER FOR COMPUTER SOLUTION!?
```

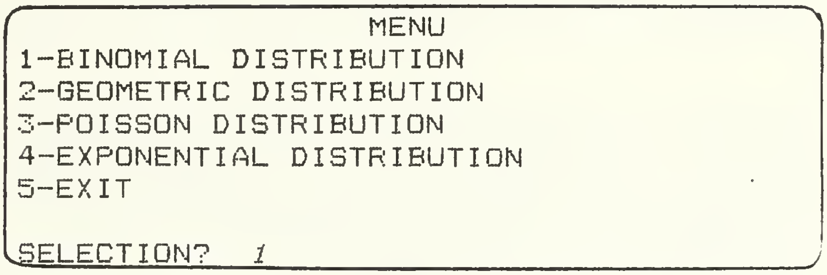
More complicated examples may require numerous repetitious calculations. In these cases, each of the calculations is displayed in a logical sequential manner building toward the final solution.



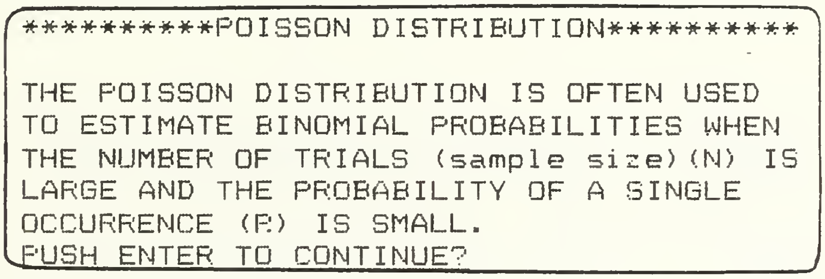
```
THE PROBABILITY FOR K = 4 IS = .0864

SELECT: 1-MENU; 2-MORE GEOMETRIC
PROBLEMS; 3-EXIT? 1
```

POISSON EXAMPLE: If a bank receives on the average $\tau=6$ bad checks per day, what is the probability that it will receive four bad checks on a given day?

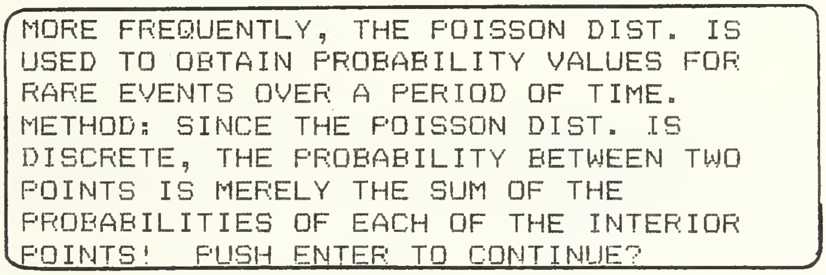
SOLUTION: Substitute $\tau=6$ and $x=4$ into the program. (Ref 2, pg 222).



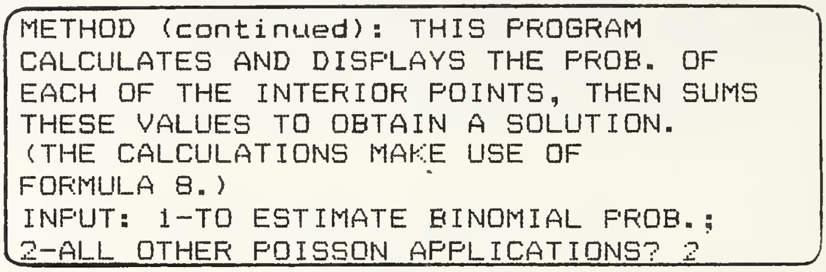
```
                    MENU
1-BINOMIAL DISTRIBUTION
2-GEOMETRIC DISTRIBUTION
3-POISSON DISTRIBUTION
4-EXPONENTIAL DISTRIBUTION
5-EXIT

SELECTION?  1
```



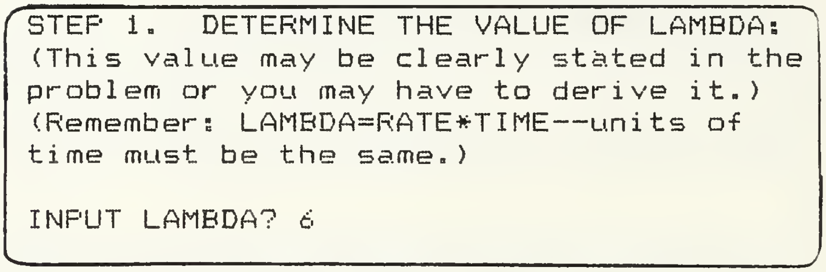
```
*********POISSON DISTRIBUTION*********

THE POISSON DISTRIBUTION IS OFTEN USED
TO ESTIMATE BINOMIAL PROBABILITIES WHEN
THE NUMBER OF TRIALS (sample size)(N) IS
LARGE AND THE PROBABILITY OF A SINGLE
OCCURRENCE (P) IS SMALL.
PUSH ENTER TO CONTINUE?
```



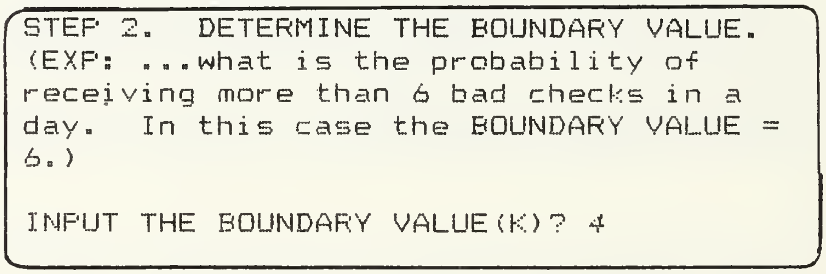
```
MORE FREQUENTLY, THE POISSON DIST. IS
USED TO OBTAIN PROBABILITY VALUES FOR
RARE EVENTS OVER A PERIOD OF TIME.
METHOD: SINCE THE POISSON DIST. IS
DISCRETE, THE PROBABILITY BETWEEN TWO
POINTS IS MERELY THE SUM OF THE
PROBABILITIES OF EACH OF THE INTERIOR
POINTS!  PUSH ENTER TO CONTINUE?
```



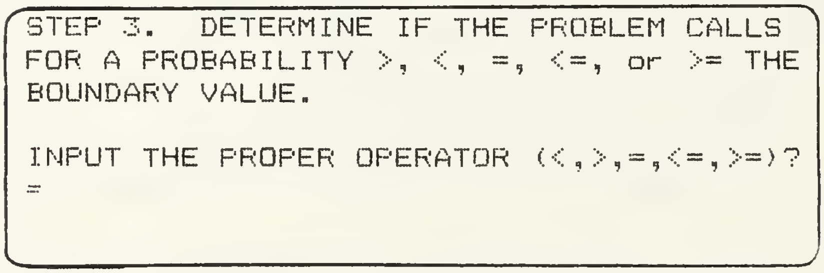
```
METHOD (continued): THIS PROGRAM
CALCULATES AND DISPLAYS THE PROB. OF
EACH OF THE INTERIOR POINTS, THEN SUMS
THESE VALUES TO OBTAIN A SOLUTION.
(THE CALCULATIONS MAKE USE OF
FORMULA 8.)
INPUT: 1-TO ESTIMATE BINOMIAL PROB.;
2-ALL OTHER POISSON APPLICATIONS? 2
```



```
STEP 1.  DETERMINE THE VALUE OF LAMBDA:
(This value may be clearly stated in the
problem or you may have to derive it.)
(Remember: LAMBDA=RATE*TIME--units of
time must be the same.)

INPUT LAMBDA? 6
```

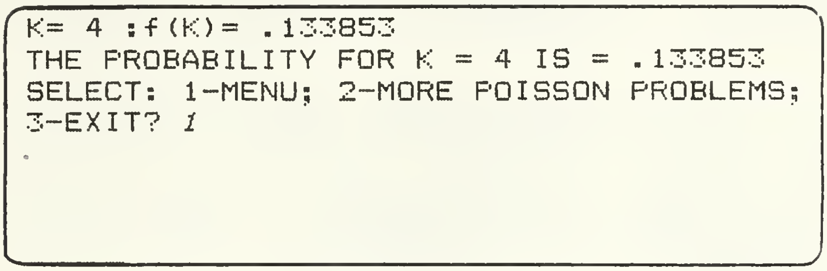


```
STEP 2.  DETERMINE THE BOUNDARY VALUE.
(EXP: ...what is the probability of
receiving more than 6 bad checks in a
day.  In this case the BOUNDARY VALUE =
6.)

INPUT THE BOUNDARY VALUE(K)? 4
```



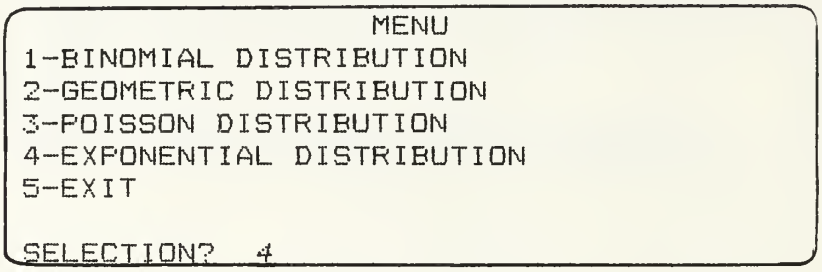
```
STEP 3.  DETERMINE IF THE PROBLEM CALLS
FOR A PROBABILITY >, <, =, <=, or >= THE
BOUNDARY VALUE.

INPUT THE PROPER OPERATOR (<,>,=,<=,>=)?
=
```

The program will check the user input for this operator and provide feedback and error correction as necessary.

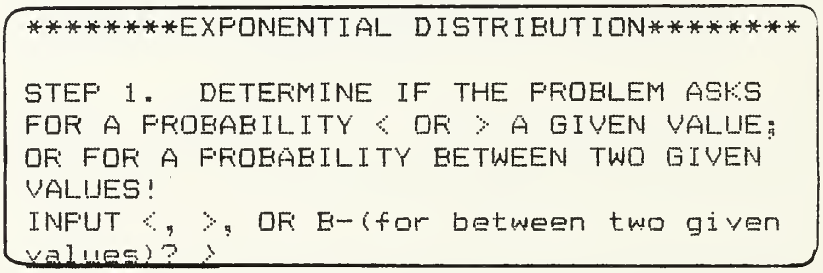


```
K= 4 :f(K)= .133853
THE PROBABILITY FOR K = 4 IS = .133853
SELECT: 1-MENU; 2-MORE POISSON PROBLEMS;
3-EXIT? 1
```

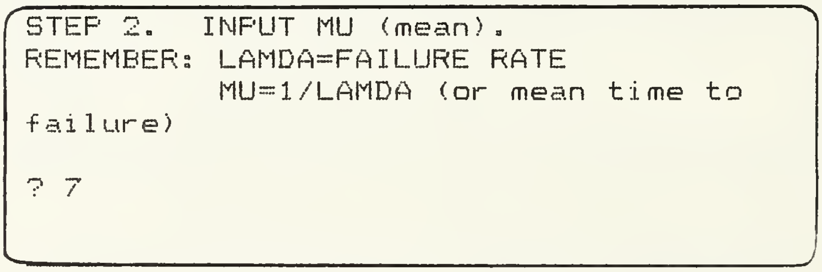
EXPONENTIAL DISTRIBUTION EXAMPLE: Given a failure rate of .1428, determine a component's reliability past 3 days.



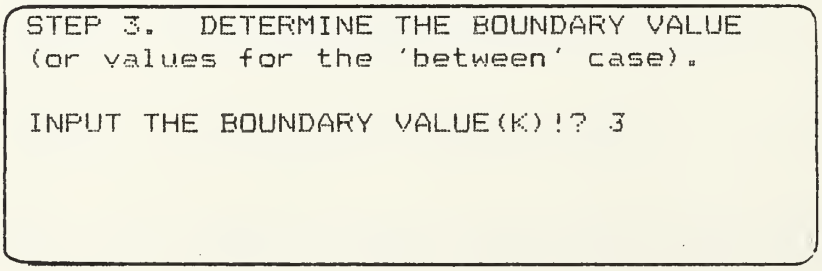
```
                    MENU
1-BINOMIAL DISTRIBUTION
2-GEOMETRIC DISTRIBUTION
3-POISSON DISTRIBUTION
4-EXPONENTIAL DISTRIBUTION
5-EXIT

SELECTION?  4
```



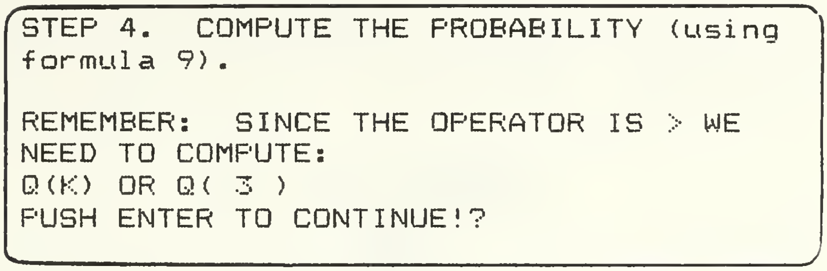
```
********EXPONENTIAL DISTRIBUTION********

STEP 1.  DETERMINE IF THE PROBLEM ASKS
FOR A PROBABILITY < OR > A GIVEN VALUE;
OR FOR A PROBABILITY BETWEEN TWO GIVEN
VALUES!
INPUT <, >, OR B-(for between two given
values)? >
```



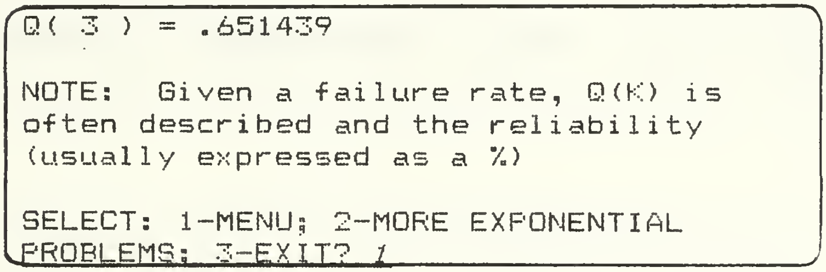
```
STEP 2.  INPUT MU (mean).
REMEMBER: LAMDA=FAILURE RATE
          MU=1/LAMDA (or mean time to
failure)

? 7
```



```
STEP 3.  DETERMINE THE BOUNDARY VALUE
(or values for the 'between' case).

INPUT THE BOUNDARY VALUE(K)!? 3
```



```
STEP 4.  COMPUTE THE PROBABILITY (using
formula 9).

REMEMBER:  SINCE THE OPERATOR IS > WE
NEED TO COMPUTE:
Q(K) OR Q( 3 )
PUSH ENTER TO CONTINUE!?
```

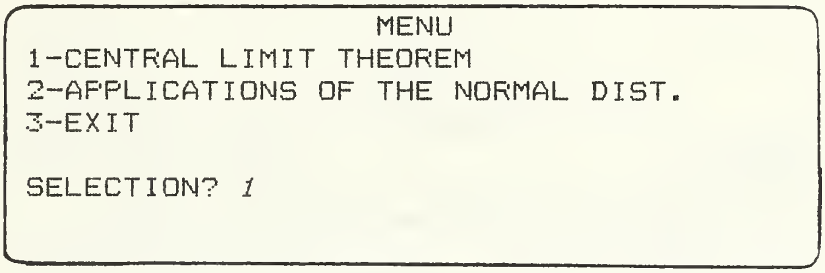


```
Q( 3 ) = .651439

NOTE:  Given a failure rate, Q(K) is
often described and the reliability
(usually expressed as a %)

SELECT: 1-MENU; 2-MORE EXPONENTIAL
PROBLEMS; 3-EXIT? 1
```

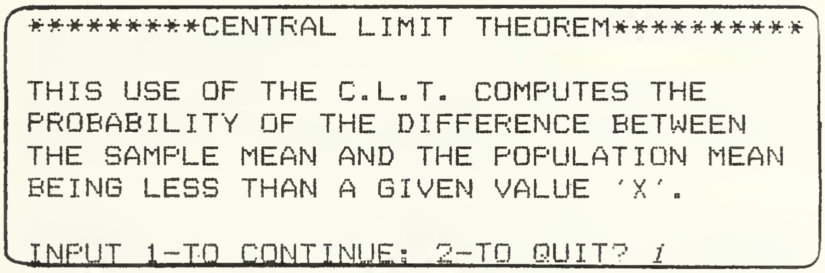
MODULE 4

EXAMPLES AND SCREEN DISPLAYS

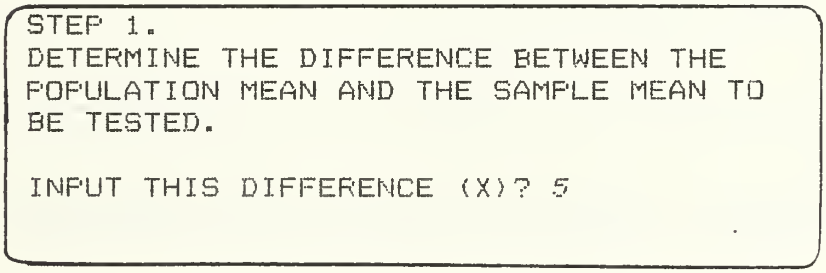
CENTRAL LIMIT THEOREM EXAMPLE: Based on the central limit theorem, what is the probability that the error will be less than 5 when we use the mean of a random sample of size n=64 to estimate the mean of an infinite population with a standard deviation of 20? (Ref 2, pg 290).



```
                    MENU
1-CENTRAL LIMIT THEOREM
2-APPLICATIONS OF THE NORMAL DIST.
3-EXIT

SELECTION? 1
```



```
*********CENTRAL LIMIT THEOREM**********

THIS USE OF THE C.L.T. COMPUTES THE
PROBABILITY OF THE DIFFERENCE BETWEEN
THE SAMPLE MEAN AND THE POPULATION MEAN
BEING LESS THAN A GIVEN VALUE 'X'.

INPUT 1-TO CONTINUE; 2-TO QUIT? 1
```



```
STEP 1.
DETERMINE THE DIFFERENCE BETWEEN THE
POPULATION MEAN AND THE SAMPLE MEAN TO
BE TESTED.

INPUT THIS DIFFERENCE (X)? 5
```

```
STEP 2.
DETERMINE FROM THE PROBLEM THE SAMPLE
SIZE AND THE POPULATION STANDARD DEV.

INPUT SAMPLE SIZE? 64
INPUT POPULATION STD. DEV.? 20
```

```
STEP 3.
USING THE VALUES OF 'X', STD. DEV. AND
THE SAMPLE SIZE, COMPUTE THE Z-SCORE
(STANDARD UNITS).  (Z-SCORE = X/(POP.
STD. DEV./SQR(SAM. SIZE))

INPUT Z-SCORE?
```

The program will check the user input for this value and provide feedback and error correction as necessary.

```
STEP 4.
USE TABLE I TO FIND THE PROPER VALUE
ASSOCIATED WITH A Z-SCORE= 2

INPUT THIS VALUE?
```

The program will check the user input for this value and provide feedback and error correction as necessary.

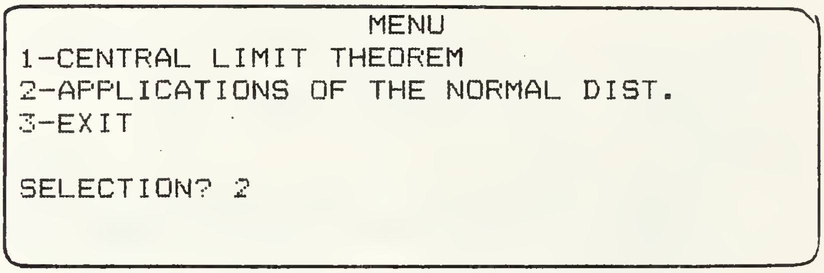
```
STEP 4.
USE TABLE I TO FIND THE PROPER VALUE
ASSOCIATED WITH A Z-SCORE= 2

INPUT THIS VALUE? 0000
WRONG TABLE VALUE!!  TRY AGAIN!
INPUT YOUR TABLE VALUE AGAIN?
```

Four zeros are input as an intentional error for demonstration purposes. The program will check the user input for this value and provide feedback and error correction as necessary.

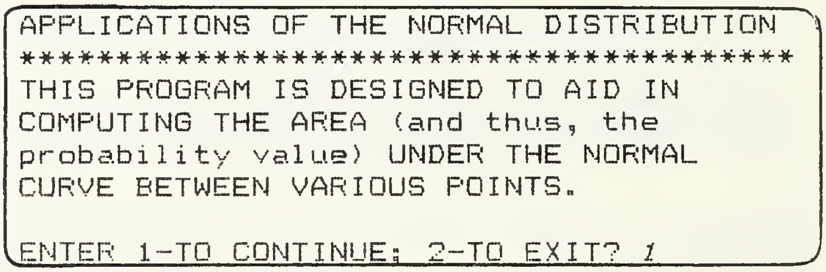
```
STEP 5.
REMEMBER, THE OBTAINED TABLE VALUE
REPRESENTS THE (PROBABILITY) AREA
BETWEEN 0 AND 2.
IN ORDER TO ACCOUNT FOR THE AREA BETWEEN
- 2 AND 0 WE MUST MULTIPLY THE TABLE
VALUE BY TWO.
PUSH ENTER TO CONTINUE?
```

```
.4772 TIMES TWO IS = .9544
THEREFORE, THIS IS THE PROBABILITY THAT
THE SAMPLE MEAN WILL DIFFER FROM THE
POP. MEAN BY LESS THAN 5.

SELECT: 1-MENU; 2-MORE C.L.T. PROBLEMS;
3-EXIT? 1
```

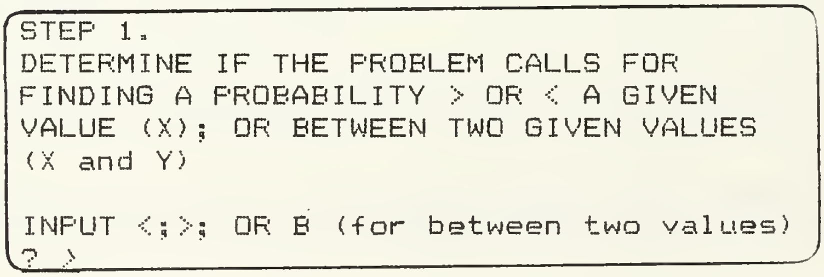
APPLICATIONS OF NORMAL DISTRIBUTION EXAMPLE: The lengths of the sardines received by a certain cannery have a mean of 4.62 inches and a standard deviation of 0.23 inch. What percentage of all these sardines are longer than 5.00 inches? (Ref 2, pg 267)



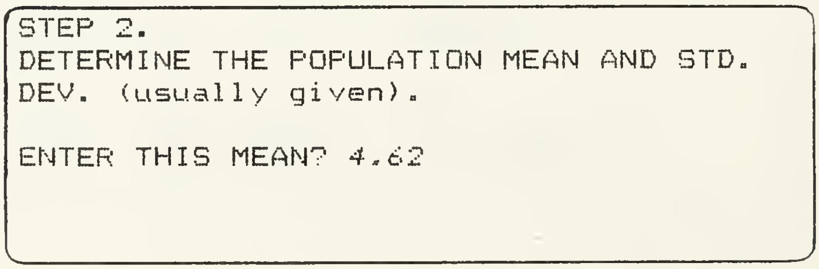
```
                    MENU
1-CENTRAL LIMIT THEOREM
2-APPLICATIONS OF THE NORMAL DIST.
3-EXIT

SELECTION? 2
```



```
APPLICATIONS OF THE NORMAL DISTRIBUTION
***************************************
THIS PROGRAM IS DESIGNED TO AID IN
COMPUTING THE AREA (and thus, the
probability value) UNDER THE NORMAL
CURVE BETWEEN VARIOUS POINTS.

ENTER 1-TO CONTINUE; 2-TO EXIT? 1
```



```
STEP 1.
DETERMINE IF THE PROBLEM CALLS FOR
FINDING A PROBABILITY > OR < A GIVEN
VALUE (X); OR BETWEEN TWO GIVEN VALUES
(X and Y)

INPUT <;>; OR B (for between two values)
? >
```



```
STEP 2.
DETERMINE THE POPULATION MEAN AND STD.
DEV. (usually given).

ENTER THIS MEAN? 4.62
```

```
STEP 2.
DETERMINE THE POPULATION MEAN AND STD.
DEV. (usually given).

ENTER THIS MEAN? 4.62
ENTER THE STD. DEV.? .23
```

```
STEP 3.
EXTRACT THE VALUE OF THE LIMIT 'X' FROM
THE PROBLEM.

ENTER LIMIT VALUE 'X'? 5
```

```
STEP 4.
CONVERT THE GIVEN LIMIT VALUE 'X' TO A
Z-SCORE.
(Z-SCORE = (X-POP.MEAN)/STD.DEV.)

INPUT THE Z-SCORE? 0000
```

Four zeros are input as an intentional error for demonstration purposes. The program will check the user input for this value and provide feedback and error correction as necessary. In this case, the error correction was flashed on the screen for a few seconds.

```
STEP 5.
FIND THE CORRESPONDING VALUE FROM TABLE
I FOR A Z-SCORE OF 1.65217.

ENTER THIS VALUE? 0000
```

Four zeros are input an an intentional error for demonstration purposes. The program will check the user input for this value and provide feedback and error correction as necessary.

```
STEP 5.
FIND THE CORRESPONDING VALUE FROM TABLE
I FOR A Z-SCORE OF 1.65217.

ENTER THIS VALUE? 0000
WRONG TABLE VALUE!! TRY AGAIN
INPUT YOUR TABLE VALUE AGAIN? 000
```

The program will check the user input for this value and provide feedback and error correction as necessary.

```
USING THE VALUE OBTAINED FROM THE TABLE,
COMPUTE THE NEEDED PROBABILITY.
REMEMBER THAT THE TABLE VALUE .4505 IS
THE AREA BETWEEN 0 AND 1.65217.  SINCE
THE OPERATOR IS >, YOU NEED TO DETERMINE
THE AREA UNDER THE CURVE TO THE RIGHT
OF 1.65217.
INPUT YOUR ANSWER!?
```

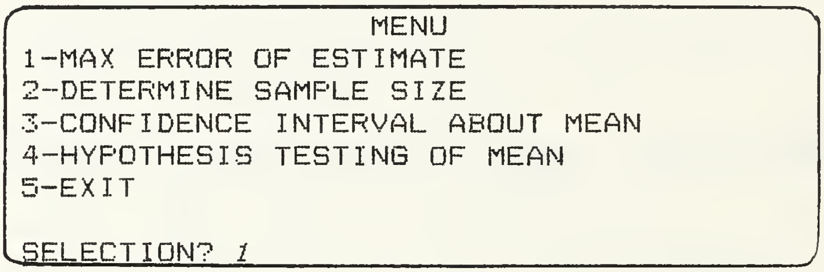
The program will check the user input for this value and provide feedback and error correction as necessary.

```
WRONG
THE PROBABILITY OF OBTAINING A VALUE >
5 IS = .0495
SELECT: 1-MENU; 2-MORE APP. OF THE
NORMAL; 3-EXIT? 1
```

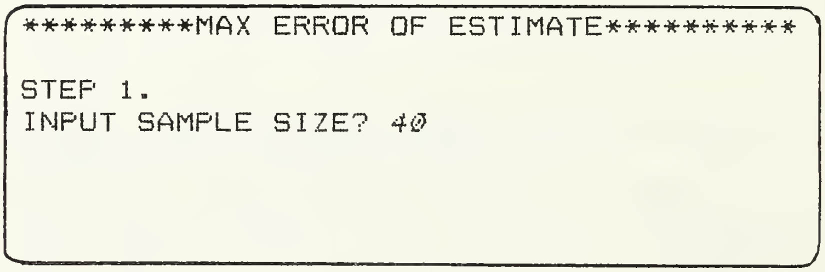
MODULE 5

EXAMPLES AND SCREEN DISPLAYS

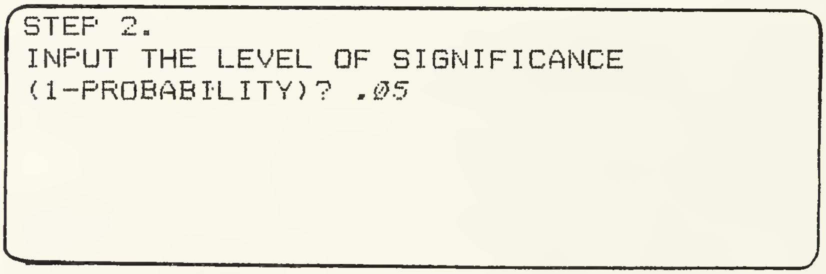
MAXIMUM ERROR OF ESTIMATE EXAMPLE: What can we assert with probability 0.95 about the maximum size of our error when we use x=19.6 tons as an estimate of a plant's true average daily emission of sulfer oxides with a sample size equal to 40? (Ref 2, pg 303)



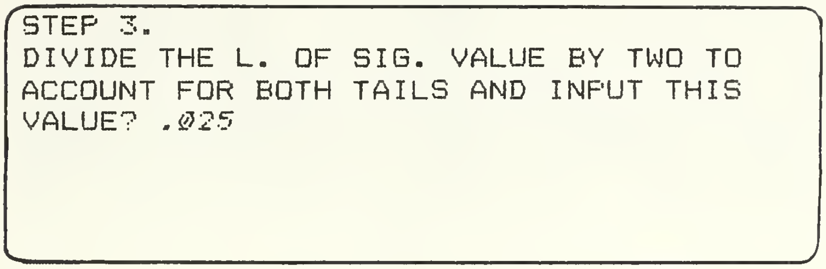
```
                   MENU
1-MAX ERROR OF ESTIMATE
2-DETERMINE SAMPLE SIZE
3-CONFIDENCE INTERVAL ABOUT MEAN
4-HYPOTHESIS TESTING OF MEAN
5-EXIT

SELECTION? 1
```



```
*********MAX ERROR OF ESTIMATE**********

STEP 1.
INPUT SAMPLE SIZE? 40
```

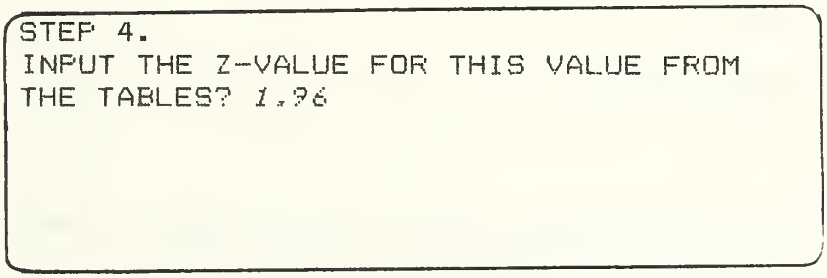


```
STEP 2.
INPUT THE LEVEL OF SIGNIFICANCE
(1-PROBABILITY)? .05
```



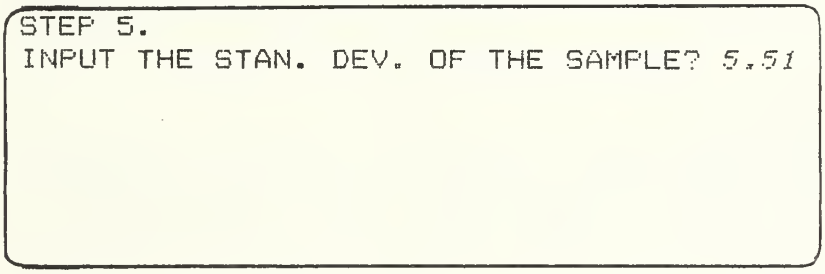
```
STEP 3.
DIVIDE THE L. OF SIG. VALUE BY TWO TO
ACCOUNT FOR BOTH TAILS AND INPUT THIS
VALUE? .025
```

The program will check the user input for this value and provide feedback and error correction as necessary.

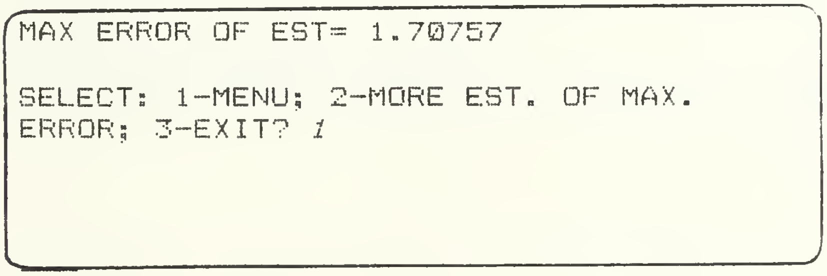


```
STEP 4.
INPUT THE Z-VALUE FOR THIS VALUE FROM
THE TABLES? 1.96
```

The program will check the user input for this value and provide feedback and error correction as necessary.

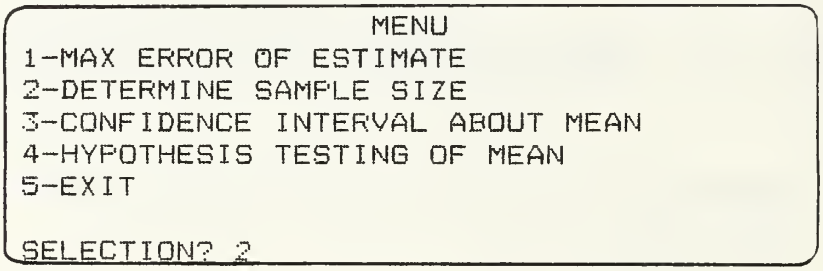


```
STEP 5.
INPUT THE STAN. DEV. OF THE SAMPLE? 5.51
```

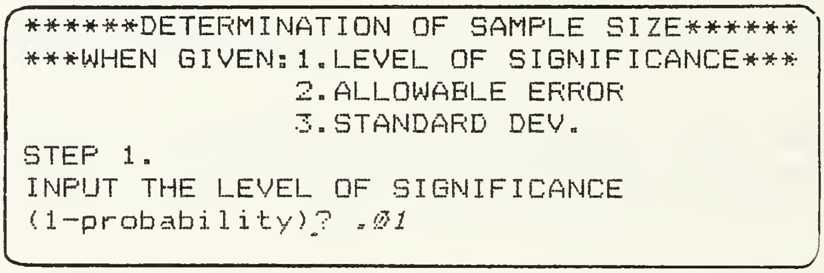


```
MAX ERROR OF EST= 1.70757

SELECT: 1-MENU; 2-MORE EST. OF MAX.
ERROR; 3-EXIT? 1
```

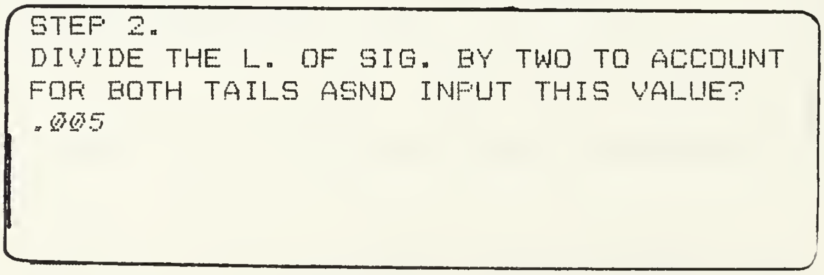
DETERMINE SAMPLE SIZE EXAMPLE: The personnel director of a manufacturing company wants to estimate the average mechanical aptitude (as measured by a certain test) of a large group of employees, and she wants this estimate to be in error by at most 2.0 with probability 0.99. If it is presumed from experience that the standard deviation equals 15.0 for this test, how large a sample is required? (Ref 2, pg 304)



```
                    MENU
1-MAX ERROR OF ESTIMATE
2-DETERMINE SAMPLE SIZE
3-CONFIDENCE INTERVAL ABOUT MEAN
4-HYPOTHESIS TESTING OF MEAN
5-EXIT

SELECTION? 2
```

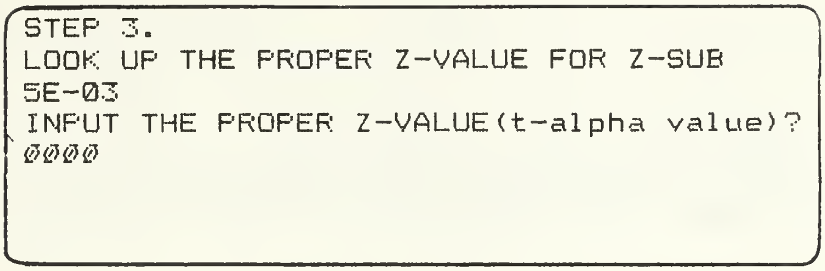


```
******DETERMINATION OF SAMPLE SIZE******
***WHEN GIVEN:1.LEVEL OF SIGNIFICANCE***
              2.ALLOWABLE ERROR
              3.STANDARD DEV.
STEP 1.
INPUT THE LEVEL OF SIGNIFICANCE
(1-probability)? .01
```



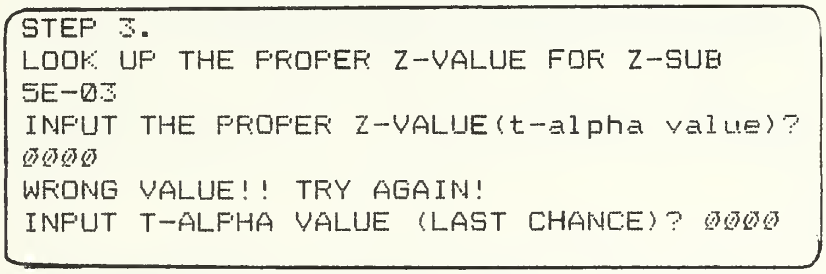
```
STEP 2.
DIVIDE THE L. OF SIG. BY TWO TO ACCOUNT
FOR BOTH TAILS ASND INPUT THIS VALUE?
.005
```

The program will check the user input for this value and provide feedback and error correction as necessary.



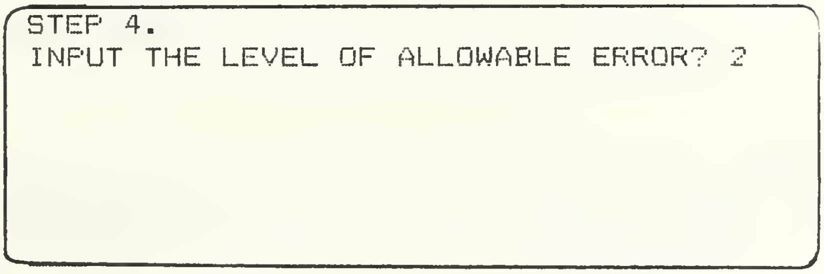
```
STEP 3.
LOOK UP THE PROPER Z-VALUE FOR Z-SUB
5E-03
INPUT THE PROPER Z-VALUE(t-alpha value)?
0000
```

Four zeros are input as an intentional error for demonstration purposes.

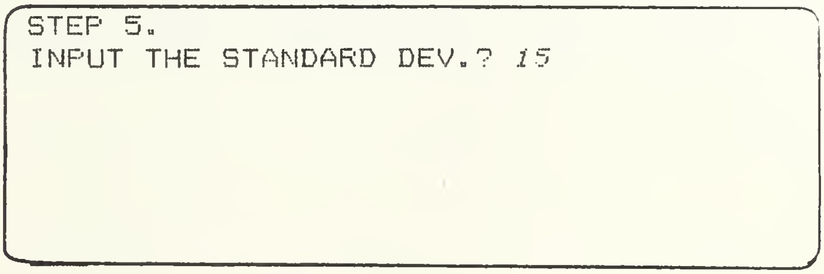


```
STEP 3.
LOOK UP THE PROPER Z-VALUE FOR Z-SUB
5E-03
INPUT THE PROPER Z-VALUE(t-alpha value)?
0000
WRONG VALUE!! TRY AGAIN!
INPUT T-ALPHA VALUE (LAST CHANCE)? 0000
```

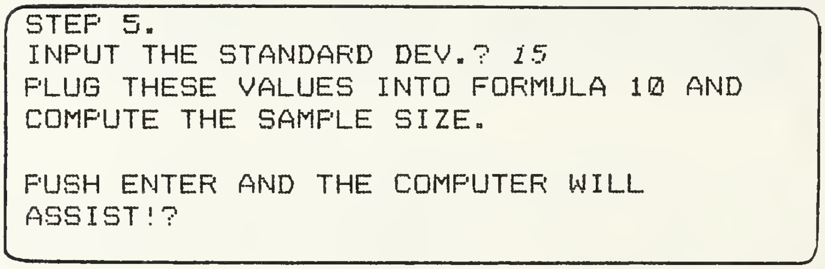
The program will check the user input for this value and provide feedback and error correction as necessary.



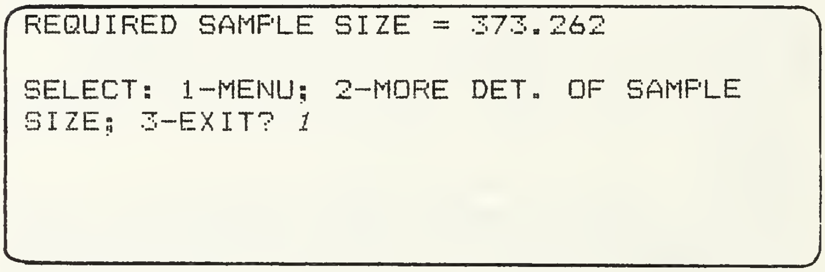
```
STEP 4.
INPUT THE LEVEL OF ALLOWABLE ERROR? 2
```



```
STEP 5.
INPUT THE STANDARD DEV.? 15
```

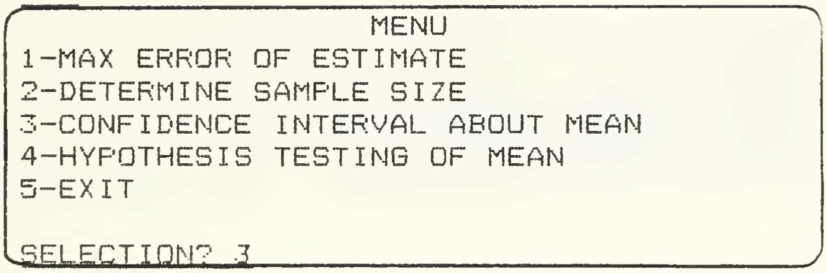


```
STEP 5.
INPUT THE STANDARD DEV.? 15
PLUG THESE VALUES INTO FORMULA 10 AND
COMPUTE THE SAMPLE SIZE.

PUSH ENTER AND THE COMPUTER WILL
ASSIST!?
```

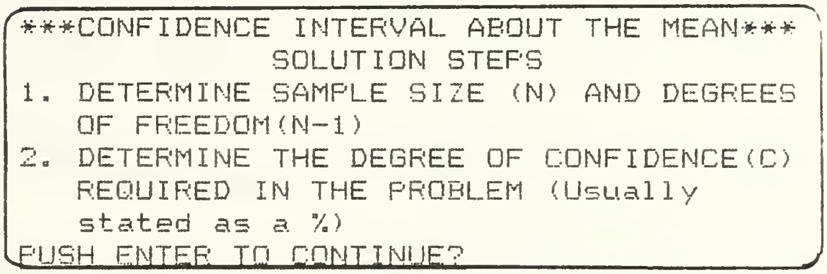


```
REQUIRED SAMPLE SIZE = 373.262

SELECT: 1-MENU; 2-MORE DET. OF SAMPLE
SIZE; 3-EXIT? 1
```

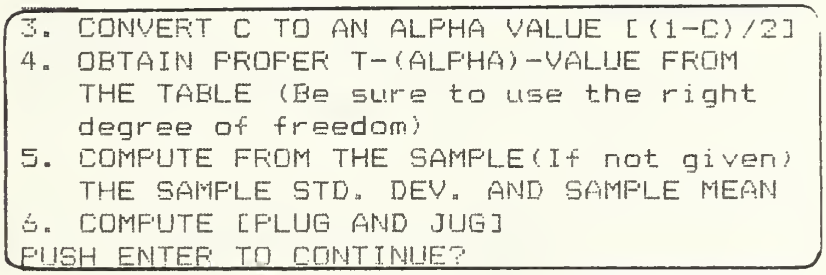
CONFIDENCE INTERVAL ABOUT THE MEAN EXAMPLE: The management of a restaurant found on the basis of a random sample size n=50 that it took its head chef on the average x-bar=14.7 minutes to prepare a particular cooked-to-order entree; the standard deviation of the sample is s=4.8 minutes. What can we assert with 90% confidence about the possible size of the error in the estimate x-bar=14.7 minutes of the true average time it takes the chef to prepare this item? (Ref 2, pg 312)



```
                   MENU
1-MAX ERROR OF ESTIMATE
2-DETERMINE SAMPLE SIZE
3-CONFIDENCE INTERVAL ABOUT MEAN
4-HYPOTHESIS TESTING OF MEAN
5-EXIT

SELECTION? 3
```



```
***CONFIDENCE INTERVAL ABOUT THE MEAN***
             SOLUTION STEPS
1. DETERMINE SAMPLE SIZE (N) AND DEGREES
   OF FREEDOM(N-1)
2. DETERMINE THE DEGREE OF CONFIDENCE(C)
   REQUIRED IN THE PROBLEM (Usually
   stated as a %)
PUSH ENTER TO CONTINUE?
```



```
3. CONVERT C TO AN ALPHA VALUE [(1-C)/2]
4. OBTAIN PROPER T-(ALPHA)-VALUE FROM
   THE TABLE (Be sure to use the right
   degree of freedom)
5. COMPUTE FROM THE SAMPLE(If not given)
   THE SAMPLE STD. DEV. AND SAMPLE MEAN
6. COMPUTE [PLUG AND JUG]
PUSH ENTER TO CONTINUE?
```

```
STEP 1.
INPUT SAMPLE SIZE? 50
```

```
STEP 1.
INPUT SAMPLE SIZE? 50
INPUT DEGREES OF FREEDOM? 48
WRONG!! TRY AGAIN!
INPUT DEGREES OF FREEDOM?
```

The program will check the user input for this value and provide feedback and error correction as necessary.

```
STEP 2.
INPUT DEGREE OF CONFIDENCE REQUIRED (in
decimal form)? .90
```

```
INPUT DEGREE OF CONFIDENCE REQUIRED (in
decimal form)? .90
STEP 3.
INPUT ALPHA VALUE (as calculated from
the deg. of conf.:(1-C)/2)? 0000
WRONG!! TRY AGAIN!
INPUT ALPHA VALUE (as calculated from
the deg. of conf.:(1-C)/2)?
```

Four zeros are input as an intentional error for demonstration purposes. The program will check the user input for this value and provide feedback and error correction as necessary.

```
STEP 4.
INPUT PROPER T-ALPHA VALUE FROM TABLE?
0000
```

Four zeros are input as an intentional error for demonstration purposes.

```
STEP 4.
INPUT PROPER T-ALPHA VALUE FROM TABLE?
0000
WRONG VALUE!! TRY AGAIN!
INPUT T-ALPHA VALUE (LAST CHANCE)? 0000
```

The program will check the user input for this value and provide feedback and error correction as necessary.

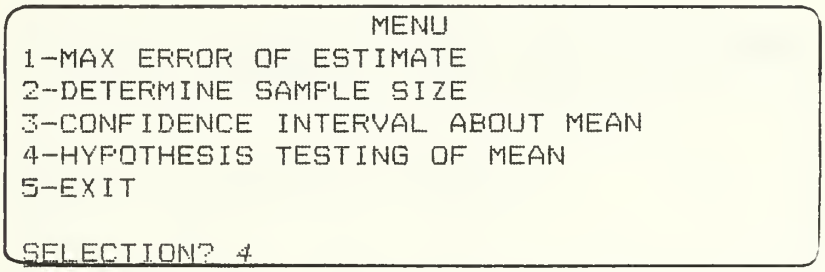
```
STEP 5.
INPUT SAMPLE STANDARD DEV.? 4.8
```

```
STEP 5.
INPUT SAMPLE STANDARD DEV.? 4.8
INPUT SAMPLE MEAN? 14.7
```

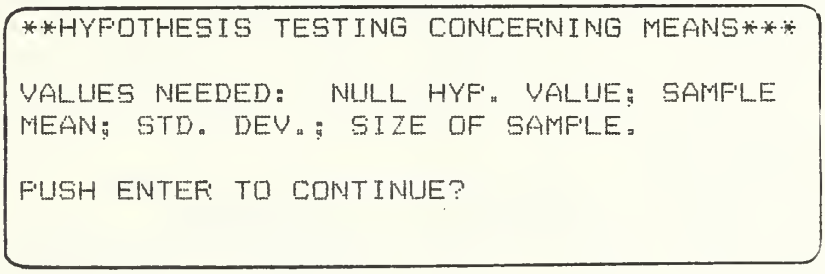
```
STEP 6.
PLUG VALUES INTO FORMULA 11 AND COMPUTE.

PUSH ENTER WHEN READY TO CONTINUE!?
```

```
THE CORRECT LOWER LIMIT= 13.5833
THE CORRECT UPPER LIMIT= 15.8167
CONFIDENCE INTERVAL =
13.5833 < TRUE MEAN < 15.8167
MAX. ERROR = 1.1166

SELECT: 1-MENU; 2-MORE C.I.; 3-EXIT? 1
```

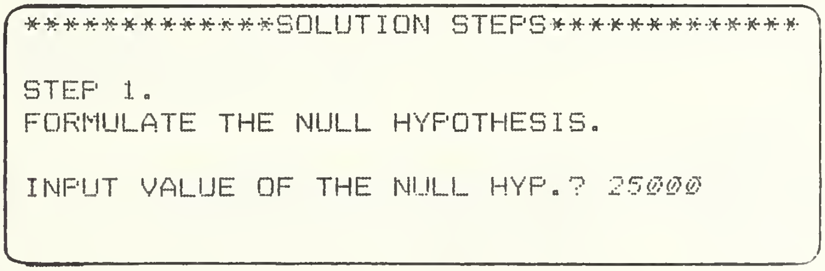
HYPOTHESIS TESTING OF THE MEAN EXAMPLE: A trucking firm suspects that the average lifetime of 25,000 miles claimed for certain tires is too high. To test the claim, the firm puts a random sample of 40 of these tires on its trucks and later finds that their mean lifetime is 24,421 miles and the standard deviation is 1,349 miles. What can it conclude at the 0.01 level of significance, if it tests the null hypothesis u=25,000 miles against the alternative hypothesis u<25,000 miles? (Ref 2, pg 330)



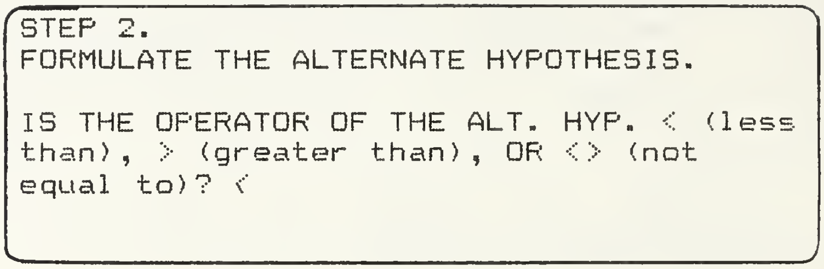
```
                    MENU
1-MAX ERROR OF ESTIMATE
2-DETERMINE SAMPLE SIZE
3-CONFIDENCE INTERVAL ABOUT MEAN
4-HYPOTHESIS TESTING OF MEAN
5-EXIT

SELECTION? 4
```



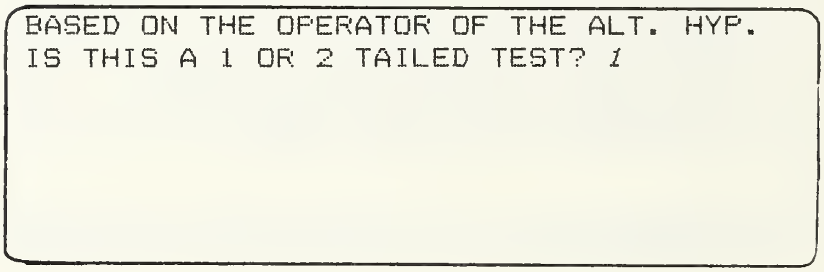
```
**HYPOTHESIS TESTING CONCERNING MEANS***

VALUES NEEDED:  NULL HYP. VALUE; SAMPLE
MEAN; STD. DEV.; SIZE OF SAMPLE.

PUSH ENTER TO CONTINUE?
```



```
************SOLUTION STEPS*************

STEP 1.
FORMULATE THE NULL HYPOTHESIS.

INPUT VALUE OF THE NULL HYP.? 25000
```

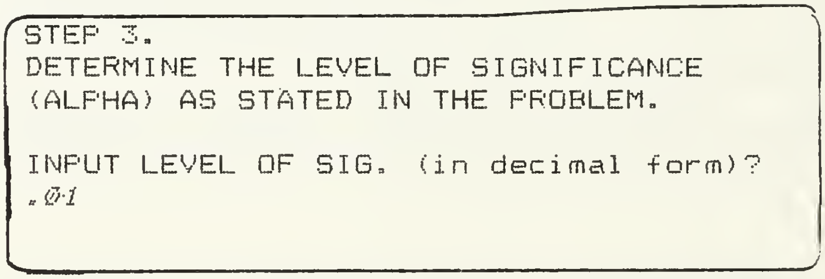


```
STEP 2.
FORMULATE THE ALTERNATE HYPOTHESIS.

IS THE OPERATOR OF THE ALT. HYP. < (less
than), > (greater than), OR <> (not
equal to)? <
```

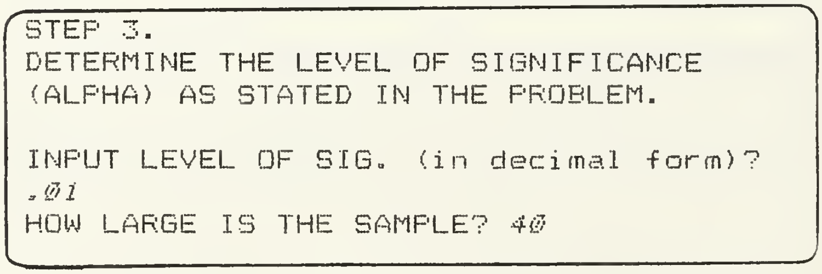


```
BASED ON THE OPERATOR OF THE ALT. HYP.
IS THIS A 1 OR 2 TAILED TEST? 1
```

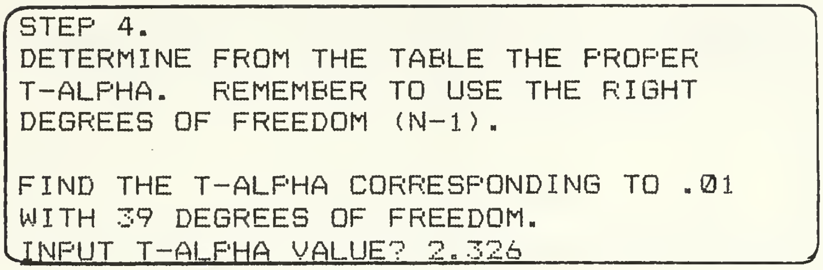
The program will check the user input for this value and provide feedback and error correction as necessary.



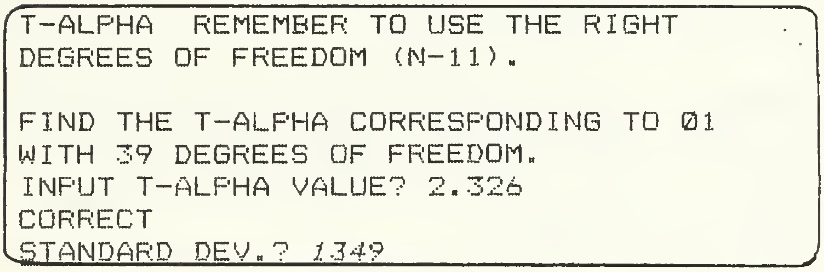
```
STEP 3.
DETERMINE THE LEVEL OF SIGNIFICANCE
(ALPHA) AS STATED IN THE PROBLEM.

INPUT LEVEL OF SIG. (in decimal form)?
.01
```



```
STEP 3.
DETERMINE THE LEVEL OF SIGNIFICANCE
(ALPHA) AS STATED IN THE PROBLEM.

INPUT LEVEL OF SIG. (in decimal form)?
.01
HOW LARGE IS THE SAMPLE? 40
```

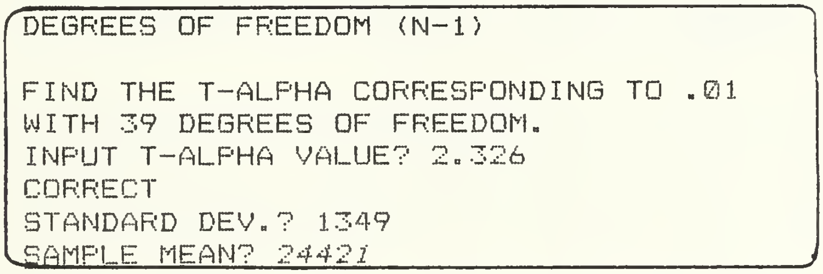


```
STEP 4.
DETERMINE FROM THE TABLE THE PROPER
T-ALPHA.  REMEMBER TO USE THE RIGHT
DEGREES OF FREEDOM (N-1).

FIND THE T-ALPHA CORRESPONDING TO .01
WITH 39 DEGREES OF FREEDOM.
INPUT T-ALPHA VALUE? 2.326
```



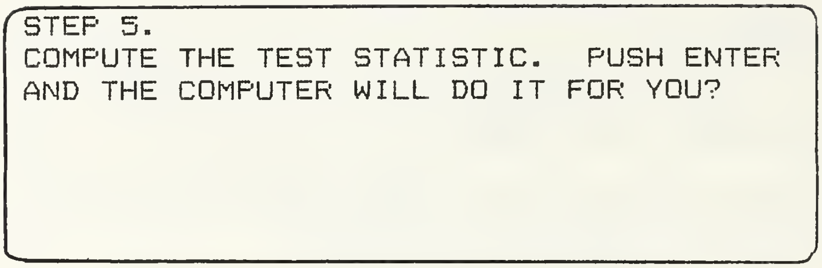
```
T-ALPHA  REMEMBER TO USE THE RIGHT
DEGREES OF FREEDOM (N-11).

FIND THE T-ALPHA CORRESPONDING TO 01
WITH 39 DEGREES OF FREEDOM.
INPUT T-ALPHA VALUE? 2.326
CORRECT
STANDARD DEV.? 1349
```

Partial screen display listings are due to the scrolling of the screen as subsequent queries appear on the screen. While this appears to be confusing in these hardcopy listings, it is hardly noticeable during actual program execution.



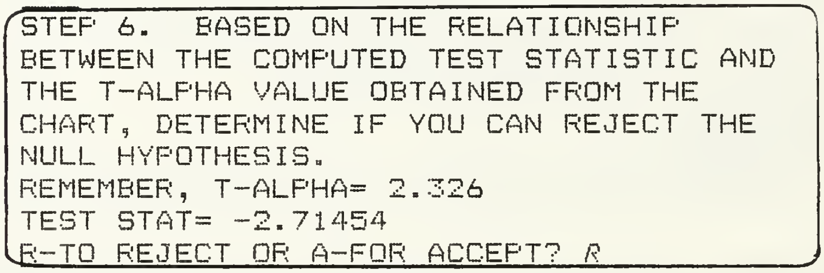
```
DEGREES OF FREEDOM (N-1)

FIND THE T-ALPHA CORRESPONDING TO .01
WITH 39 DEGREES OF FREEDOM.
INPUT T-ALPHA VALUE? 2.326
CORRECT
STANDARD DEV.? 1349
SAMPLE MEAN? 24421
```

Partial screen display listings are due to the scrolling of the screen as subsequent queries appear on the screen. While this appears to be confusing in these hardcopy listings, it is hardly noticeable during actual program execution.

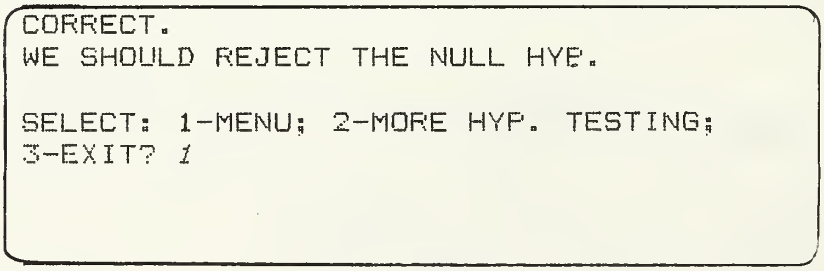


```
STEP 5.
COMPUTE THE TEST STATISTIC.  PUSH ENTER
AND THE COMPUTER WILL DO IT FOR YOU?
```

The program will briefly display all the input values again for user perusal and then compute the test statistic.



```
STEP 6.  BASED ON THE RELATIONSHIP
BETWEEN THE COMPUTED TEST STATISTIC AND
THE T-ALPHA VALUE OBTAINED FROM THE
CHART, DETERMINE IF YOU CAN REJECT THE
NULL HYPOTHESIS.
REMEMBER, T-ALPHA= 2.326
TEST STAT= -2.71454
R-TO REJECT OR A-FOR ACCEPT? R
```

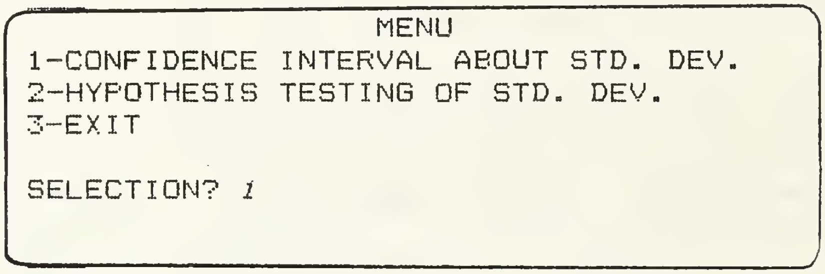


```
CORRECT.
WE SHOULD REJECT THE NULL HYP.

SELECT: 1-MENU; 2-MORE HYP. TESTING;
3-EXIT? 1
```

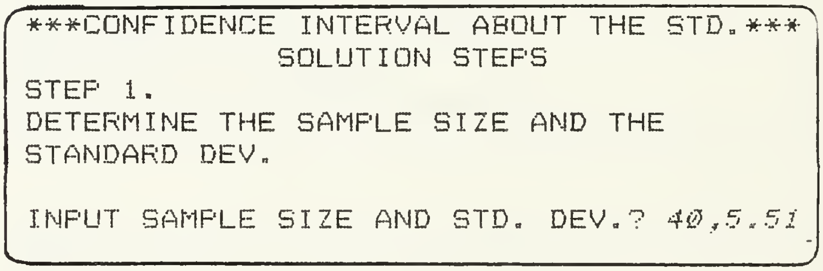
MODULE 6

EXAMPLES AND SCREEN DISPLAYS

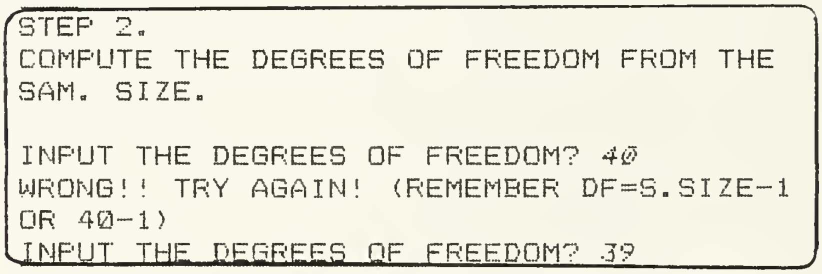
CONFIDENCE INTERVAL ABOUT THE STANDARD DEVIATION EXAMPLE: If s=5.51 tons for a large industrial plant's emission of sulfur oxides on n=40 days, construct a 95% confidence interval for the standard deviation of the population sampled. (Ref 2, pg 348)



```
                    MENU
1-CONFIDENCE INTERVAL ABOUT STD. DEV.
2-HYPOTHESIS TESTING OF STD. DEV.
3-EXIT

SELECTION? 1
```



```
***CONFIDENCE INTERVAL ABOUT THE STD.***
             SOLUTION STEPS
STEP 1.
DETERMINE THE SAMPLE SIZE AND THE
STANDARD DEV.

INPUT SAMPLE SIZE AND STD. DEV.? 40,5.51
```



```
STEP 2.
COMPUTE THE DEGREES OF FREEDOM FROM THE
SAM. SIZE.

INPUT THE DEGREES OF FREEDOM? 40
WRONG!! TRY AGAIN! (REMEMBER DF=S.SIZE-1
OR 40-1)
INPUT THE DEGREES OF FREEDOM? 39
```

The program will check the user input for this value and provide feedback and error correction as necessary.

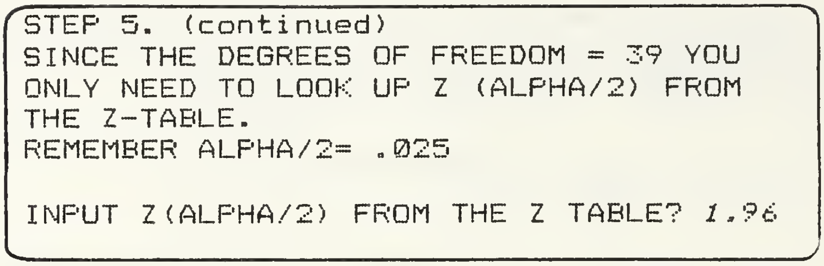
```
STEP 3.
DETERMINE THE LEVEL OF CONFIDENCE
REQUIRED.

INPUT THE L. OF CONFIDENCE (C) IN
DECIMAL FORM? .95
```

```
STEP 4.
CONVERT C TO AN ALPHA VALUE (1-C).

INPUT ALPHA VALUE? .005
WRONG!! TRY AGAIN!
CONVERT C TO AN ALPHA VALUE (1-C).

INPUT ALPHA VALUE? .05
```

The program will check the user input for this value and provide feedback and error correction as necessary.

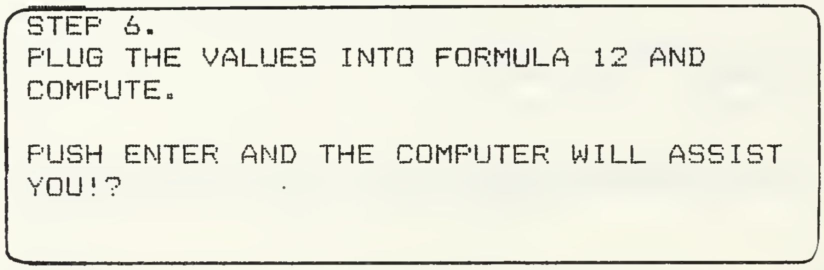
```
STEP 5.
IF THE DEGREES OF FREEDOM IS > 30 THEN
USE THE 'Z' TABLE TO FIND A VALUE FOR
Z (ALPHA/2).
IF THE D.O.F. IS < 30 THEN YOU MUST
DETERMINE TWO VALUES FROM THE CHI-SQUARE
'X' TABLE: X(ALPHA/2) and X(1-ALPHA/2)
WHICH TABLE DO YOU NEED? Z or X? Z
```

The program will check the user input for this value and provide feedback and error correction as necessary.

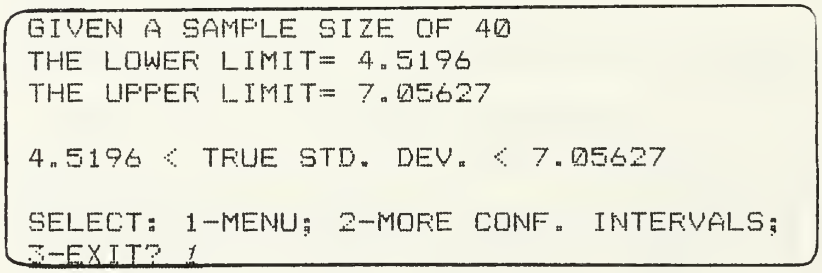


```
STEP 5. (continued)
SINCE THE DEGREES OF FREEDOM = 39 YOU
ONLY NEED TO LOOK UP Z (ALPHA/2) FROM
THE Z-TABLE.
REMEMBER ALPHA/2= .025

INPUT Z(ALPHA/2) FROM THE Z TABLE? 1.96
```

The program will check the user input for this value and provide feedback and error correction as necessary.

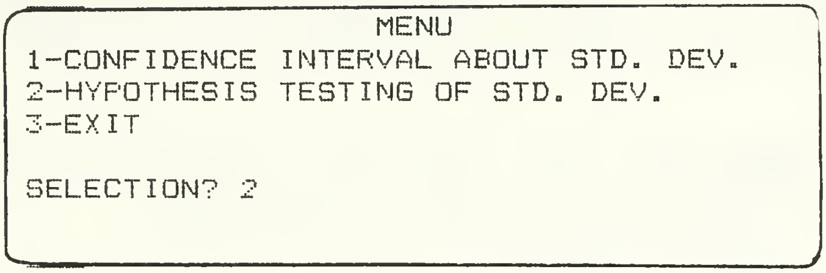


```
STEP 6.
PLUG THE VALUES INTO FORMULA 12 AND
COMPUTE.

PUSH ENTER AND THE COMPUTER WILL ASSIST
YOU!?
```

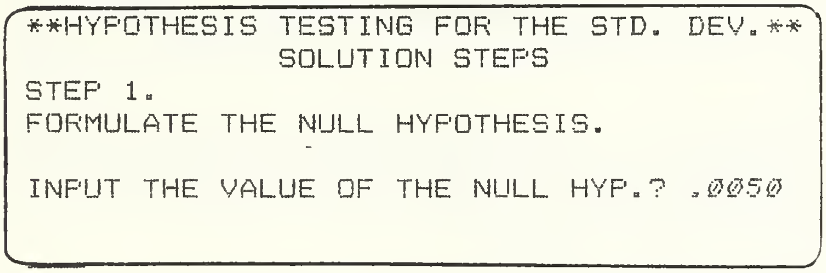


```
GIVEN A SAMPLE SIZE OF 40
THE LOWER LIMIT= 4.5196
THE UPPER LIMIT= 7.05627

4.5196 < TRUE STD. DEV. < 7.05627

SELECT: 1-MENU; 2-MORE CONF. INTERVALS;
3-EXIT? 1
```

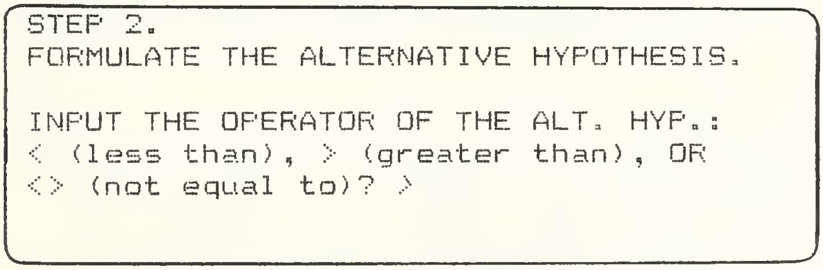
HYPOTHESIS TESTING OF THE STANDARD DEVIATION EXAMPLE: Specifications for mass-produced bearings of a certain type require, among other things, that the standard deviation of their outside diameters should not exceed 0.0050 cm. Use the level of significance 0.01 to test the null hypothesis $\sigma = 0.0050$ against the alternative hypothesis $\sigma > 0.0050$ on the basis of a random sample of size $n = 12$ for which $s = 0.0077$ cm. (Ref 2, pg 351)



```
                    MENU
1-CONFIDENCE INTERVAL ABOUT STD. DEV.
2-HYPOTHESIS TESTING OF STD. DEV.
3-EXIT

SELECTION? 2
```



```
**HYPOTHESIS TESTING FOR THE STD. DEV.**
              SOLUTION STEPS
STEP 1.
FORMULATE THE NULL HYPOTHESIS.

INPUT THE VALUE OF THE NULL HYP.? .0050
```



```
STEP 2.
FORMULATE THE ALTERNATIVE HYPOTHESIS.

INPUT THE OPERATOR OF THE ALT. HYP.:
< (less than), > (greater than), OR
<> (not equal to)? >
```

```
BASED ON THE OPERATOR OF THE ALT. HYP.
IS THIS A 1 OR 2 TAILED TEST? 1
```

The program will check the user input for this value and provide feedback and error correction as necessary.

```
STEP 3.
DETERMINE THE LEVEL OF SIGNIFICANCE
(alpha) AS STATED IN THE PROBLEM.

INPUT LEVEL OF SIG.? .01
```

```
STEP 4.
EXTRACT SAMPLE SIZE AND STANDARD DEV.
FROM THE PROBLEM.

INPUT SAMPLE SIZE, STD. DEV.? 12
?? .0077
```

```
STEP 5.
DETERMINE THE DEGREES OF FREEDOM FROM
THE SAMPLE SIZE (SAM. SZ.-1 OR 12 -1).

INPUT THE DEGREES OF FREEDOM? 11
```

The program will check the user input for this value and provide feedback and error correction as necessary.

```
STEP 6.
IF THE DEGREES OF FREEDOM ARE > THAN 30
THEN YOU ONLY NEED TO FIND ONE VALUE
FROM THE Z-TABLE.  IF THE D.O.F. ARE <
30 THEN YOU NEED TO FIND THE VALUE(S)
FROM THE CHI-SQUARE (X) TABLE.
WHICH TABLE DO YOU NEED-Z OR X? X
```

The program will check the user input for this value and provide feedback and error correction as necessary.

```
STEP 6. (Continued)
SINCE THE DEGREES OF FREEDOM = 11 AND
THE SIGN OF THE ALT. HYP. IS > YOU WILL
ONLY BE REQUIRED TO LOOK UP ONE VALUE
IN THE CHI-SQUARE (X) TABLE!!

PUSH ENTER TO CONTINUE?
```

```
IN THIS CASE (>) FIND THE X(alpha) FROM
THE CHI-SQUARE TABLE.
INPUT THE VALUE FOR 11 DEGREES OF
FREEDOM, FOR X( .01 )? 24.725
```

Due to the variable lengths of some of the input values, some screen displays will experience "word chop" or the seemingly random division of words at the end of a line. This is caused by the inability of the programming language to dynamically adjust the output.

```
STEP 7.
THE NEXT STEP IS TO COMPUTE THE TEST
STATISTIC (FORMULA 13).

PUSH ENTER AND THE COMPUTER WILL
ASSIST!?
```

The program will briefly display all the input values again for user perusal and then compute the test statistic.

```
STEP 8.
BASED ON THE RELATIONSHIP BETWEEN THE
COMPUTED TEST STATISTIC AND THE VALUE(S)
OBTAINED FROM THE TABLE, DETERMINE IF
YOU CAN REJECT THE NULL HYPOTHESIS.

PUSH ENTER TO CONTINUE?
```

```
REMEMBER:
THE TEST STAT.= 26.0876
THE TABLE VALUE= 24.725
INPUT: R-TO REJECT OR A-TO ACCEPT? R
```

The program will check the user input for this value and provide feedback and error correction as necessary.

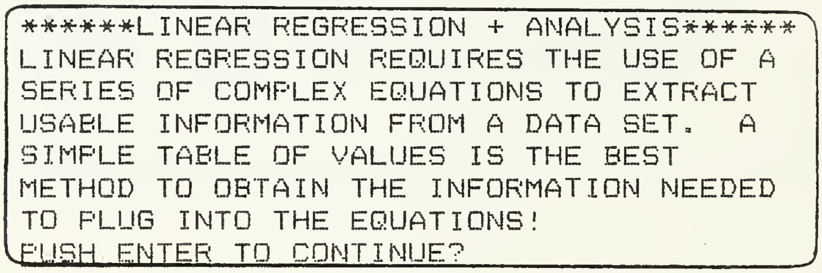
```
CORRECT.
WE SHOULD REJECT THE NULL HYPOTHESIS.

SELECT: 1-MENU; 2-MORE HYP. TESTING;
3-EXIT? 1
```

MODULE 7

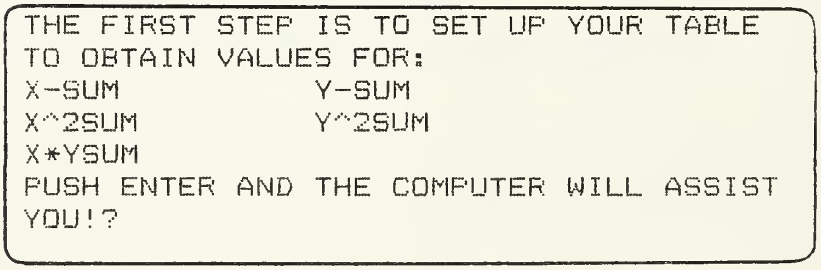
EXAMPLES AND SCREEN DISPLAYS

MASTER DATA SET FOR MODULE 7 EXAMPLES

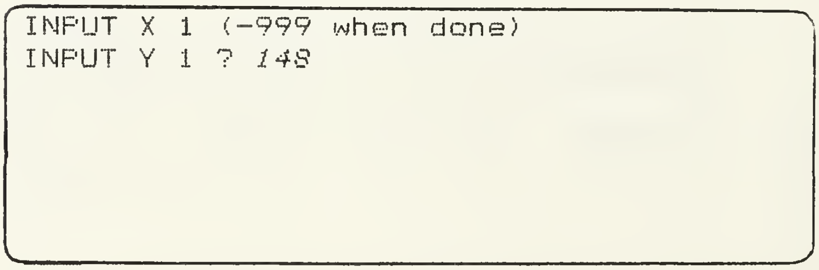
Year X	Maintenance Cost (Dollars) Y
4	148
2	128
3	133
5	154
2	118
3	145
4	143
5	159
4	142
3	127
35	1,397



```
******LINEAR REGRESSION + ANALYSIS******
LINEAR REGRESSION REQUIRES THE USE OF A
SERIES OF COMPLEX EQUATIONS TO EXTRACT
USABLE INFORMATION FROM A DATA SET.  A
SIMPLE TABLE OF VALUES IS THE BEST
METHOD TO OBTAIN THE INFORMATION NEEDED
TO PLUG INTO THE EQUATIONS!
PUSH ENTER TO CONTINUE?
```



```
THE FIRST STEP IS TO SET UP YOUR TABLE
TO OBTAIN VALUES FOR:
X-SUM               Y-SUM
X^2SUM              Y^2SUM
X*YSUM
PUSH ENTER AND THE COMPUTER WILL ASSIST
YOU!?
```



```
INPUT X 1 (-999 when done)
INPUT Y 1 ? 148
```

The user continues to input all data pairs until completed. The program will then summarize the input data and provide a formula for the regression line. Following this, an options menu will be displayed.

```
INPUT X 2 (-999 when done)?

   X           Y
   4           148

   X^2         Y^2          X*Y
   16          21904        592
```

```
INPUT X 11 (-999 when done)? -999

   X           Y
   3           127

   X^2         Y^2           X*Y
   9           16129         381
```

```
X-SUM= 35      Y-SUM= 1397
Y^2SUM= 133
Y^2SUM= 196665
X*YSUM= 5004
LEAST-SQUARES LINE:Y-hat= 101.533 +
10.9048 X
WHERE a= 101.533 AND b= 10.9048
PUSH ENTER TO CONTINUE?
```

HYPOTHESIS TESTING OF ALPHA AND BETA EXAMPLE: With the master data set, use the 0.05 level of significance to test the null hypothesis alpha=$125.00 against the alternative hypothesis alpha<$125,000. (Ref 2, pg 446)

```
                         MENU
1-HYP. TEST
2-CONF. INT. (for alpha or beta)
3-CONF. INT. (mean of Y at Xo)
4-LIMITS OF PRED.
5-ESTIMATES OF y WHEN GIVEN x
6-EXIT
SELECTION? 1
```

```
**********HYPOTHESIS TESTING**********

WOULD YOU LIKE TO TEST alpha OR beta?
A-FOR alpha; B-FOR beta? A
```

```
************SOLUTION STEPS************

STEP 1.  FORMULATE THE NULL HYPOTHESIS.

INPUT VALUE OF THE NULL HYP.? 125
```

```
STEP 2.  FORMULATE THE ALTERNATE
HYPOTHESIS.

IS THE OPERATOR OF THE ALT. HYP. < (less
than), > (greater than) OR <>(not equal
to)? <
```

```
BASED ON THE OPERATOR OF THE ALT. HYP.
IS THIS A 1 OR 2 TAILED TEST? 1
```

The program will check the user input for this value and provide feedback and error correction as necessary.

```
STEP 3.  DETERMINE THE LEVEL OF
SIGNIFICANCE (ALPHA) AS STATED IN THE
PROBLEM.

INPUT LEVEL OF SIG. (always a decimal
value)? .05
```

```
STEP 4.  DETERMINE FROM THE TABLE THE
PROPER T-ALPHA.  REMEMBER TO USE THE
RIGHT DEGREES OF FREEDOM (N-2).

FIND THE T-ALPHA CORRESPONDING TO .05
WITH 8 DEGREES OF FREEDOM.
INPUT T-ALPHA VALUE? 1.86
```

The program will check the user input for this value and provide feedback and error correction as necessary.

```
STEP 5.  THE NEXT STEP IS TO COMPUTE
THE TEST STATISTIC (FORMULA 15).

PUSH ENTER AND THE COMPUTER WILL DO IT
FOR YOU?
```

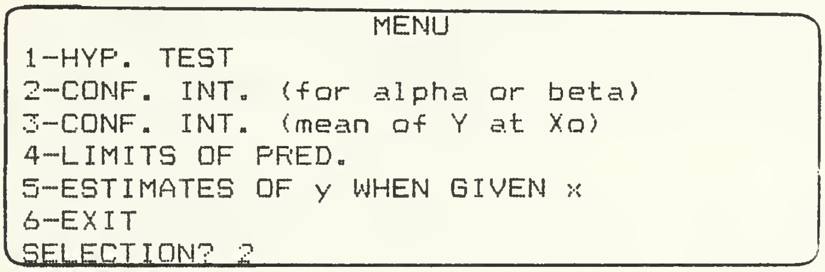
The program will briefly display all the input values again for user perusal and then compute the test statistic.

```
STEP 6.  BASED ON THE RELATIONSHIP
BETWEEN THE COMPUTED TEST STATISTIC
AND THE T-ALPHA VALUE OBTAINED FROM THE
CHART, DETERMINE IF YOU CAN REJECT THE
NULL HYPOTHESIS.  REMEMBER, T-ALPHA=
1.86; TEST STAT= -3.68755
R-TO REJECT OR A-TO ACCEPT? R
```

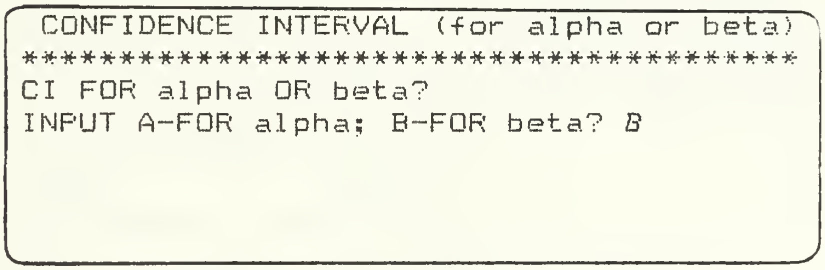
The program will check the user input for this value and provide feedback and error correction as necessary.

```
CORRECT.  WE SHOULD REJECT THE NULL
HYP.

SELECT: 1-MENU; 2-MORE HYP. TESTING;
3-ENTER NEW DATA SET; 4-EXIT? 1
```

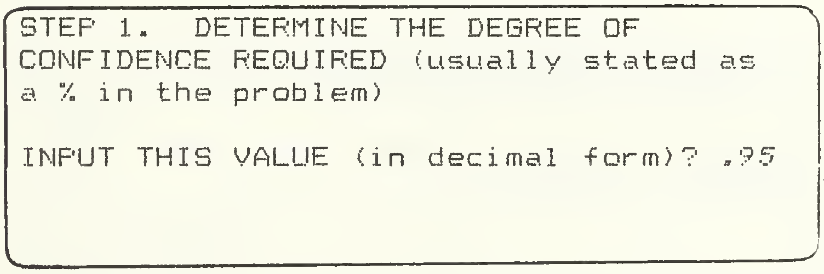
CONFIDENCE INTERVAL FOR ALPHA OR BETA EXAMPLE: Using the master data set, construct a 95% confidence interval about beta.



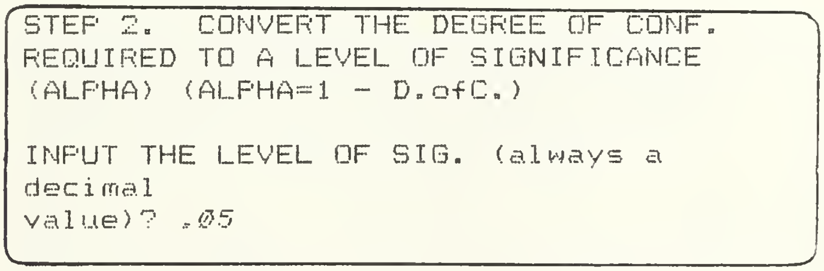
```
                    MENU
1-HYP. TEST
2-CONF. INT. (for alpha or beta)
3-CONF. INT. (mean of Y at Xo)
4-LIMITS OF PRED.
5-ESTIMATES OF y WHEN GIVEN x
6-EXIT
SELECTION? 2
```



```
 CONFIDENCE INTERVAL (for alpha or beta)
****************************************
CI FOR alpha OR beta?
INPUT A-FOR alpha; B-FOR beta? B
```



```
STEP 1.  DETERMINE THE DEGREE OF
CONFIDENCE REQUIRED (usually stated as
a % in the problem)

INPUT THIS VALUE (in decimal form)? .95
```



```
STEP 2.  CONVERT THE DEGREE OF CONF.
REQUIRED TO A LEVEL OF SIGNIFICANCE
(ALPHA) (ALPHA=1 - D.ofC.)

INPUT THE LEVEL OF SIG. (always a
decimal
value)? .05
```

```
STEP 3.  USING TABLE II, FIND THE PROPER
T-ALPHA VALUE.  (Remember the degrees of
freedom=(n-2): in this case DFRD= 8)
(Since all CIs are 2 tailed, divide
the level of sig. by 2: in this case
LSIG= .025)
ENTER THE TABLE VALUE (T-ALPHA)? 2.306
```

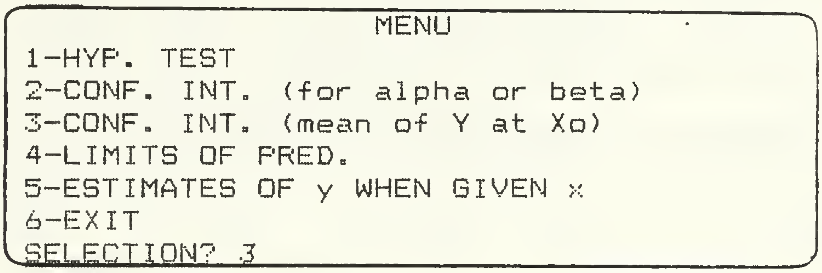
The program will check the user input for this value and provide feedback and error correction as necessary.

```
COMPUTE THE CI INTERVAL (using
formula 16).

PUSH ENTER AND THE COMPUTER WILL DO IT
FOR YOU?
```

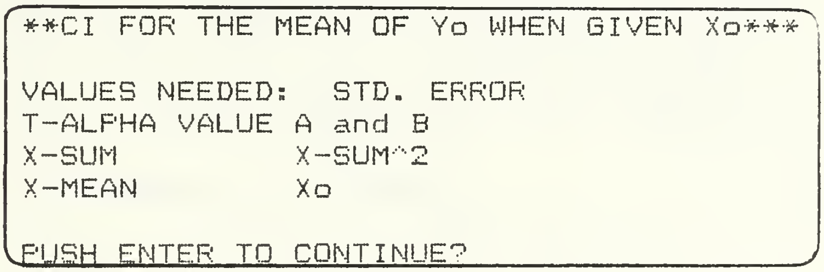
The program will briefly display all the input values again for user perusal and then compute the test statistic.

```
6.88085 < BETA < 14.9288

SELECT: 1-MENU; 2-MORE CIs; 3-NEW DATA
ENTRY; 4-EXIT? 1
```

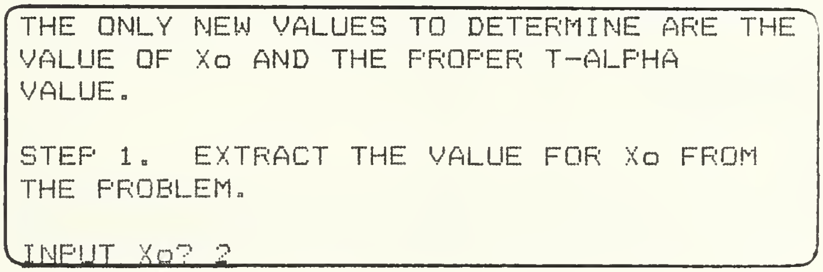
CONFIDENCE INTERVAL FOR THE MEAN OF Y AT X NAUGHT EXAMPLE: Using the master data set, find a 95% confidence interval for the true average cost of maintenance during the second year. (Ref 2, pg 447)



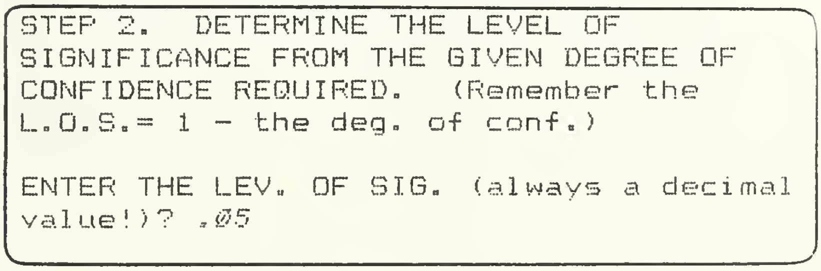
```
                          MENU
1-HYP. TEST
2-CONF. INT. (for alpha or beta)
3-CONF. INT. (mean of Y at Xo)
4-LIMITS OF PRED.
5-ESTIMATES OF y WHEN GIVEN x
6-EXIT
SELECTION? 3
```



```
**CI FOR THE MEAN OF Yo WHEN GIVEN Xo***

VALUES NEEDED:  STD. ERROR
T-ALPHA VALUE A and B
X-SUM          X-SUM^2
X-MEAN         Xo

PUSH ENTER TO CONTINUE?
```



```
THE ONLY NEW VALUES TO DETERMINE ARE THE
VALUE OF Xo AND THE PROPER T-ALPHA
VALUE.

STEP 1.  EXTRACT THE VALUE FOR Xo FROM
THE PROBLEM.

INPUT Xo? 2
```



```
STEP 2.  DETERMINE THE LEVEL OF
SIGNIFICANCE FROM THE GIVEN DEGREE OF
CONFIDENCE REQUIRED.  (Remember the
L.O.S.= 1 - the deg. of conf.)

ENTER THE LEV. OF SIG. (always a decimal
value!)? .05
```

```
STEP 3.  USING TABLE II, FIND THE PROPER
T-ALPHA VALUE.  (Remember the degrees of
freedom=(N-2): in this case DRFD= 8)
(Also, since all CIs are 2 tailed tests,
divide your level of sig. by two: in
this case L.O.S.= .025)

ENTER THE TABLE VALUE T-ALPHA? 2.306
```

The program will check the user input for this value and provide feedback and error correction as necessary.

```
STEP 4.  COMPUTE THE CONFIDENCE INTERVAL
USING FORMULA 17.

PUSH ENTER AND THE COMPUTER WILL
ASSIST!?
```

```
116.033 < Yo < 130.652

SELECT: 1-MENU; 2-NEW DATA ENTRY;
3-EXIT? 1
```

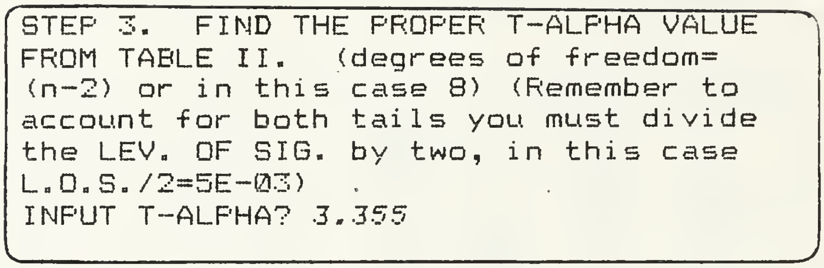
LIMITS OF PREDICTIONS EXAMPLE: Develop the prediction limits for the master data set at the 99% level for x=2.

```
                    MENU
1-HYP. TEST
2-CONF. INT. (for alpha or beta)
3-CONF. INT. (mean of Y at Xo)
4-LIMITS OF PRED.
5-ESTIMATES OF y WHEN GIVEN x
6-EXIT
SELECTION? 4
```

```
**********LIMITS OF PREDICTION**********

THE COMPUTATIONAL STEPS ARE IDENTICAL TO
THE DEVELOPMENT OF A CI ABOUT THE MEAN
OF Y.  (In this case, however, formula
18 is used.)

PUSH ENTER TO CONTINUE?
```

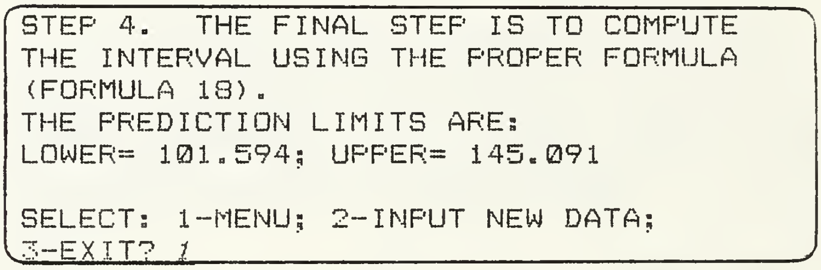
```
STEP 1.  DETERMINE Xo FROM THE PROBLEM.

INPUT Xo? 2
```

```
STEP 2.  DETERMINE THE LEVEL OF
SIGNIFICANCE NEEDED.  (Remember L.O.S.=
(1 - degree of confidence)

INPUT THE LEV. OF SIG. (always a decimal
value!)? .01
```



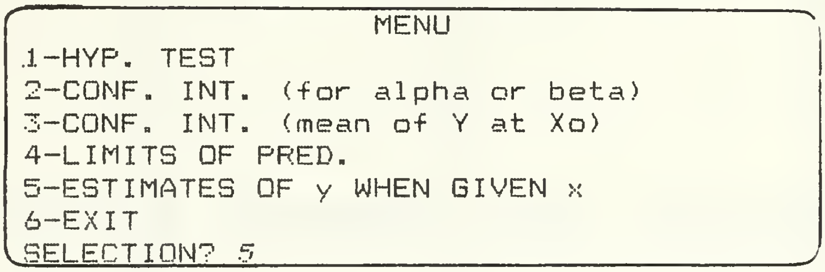
```
STEP 3.  FIND THE PROPER T-ALPHA VALUE
FROM TABLE II.  (degrees of freedom=
(n-2) or in this case 8) (Remember to
account for both tails you must divide
the LEV. OF SIG. by two, in this case
L.O.S./2=5E-03)
INPUT T-ALPHA? 3.355
```

The program will check the user input for this value and provide feedback and error correction as necessary.

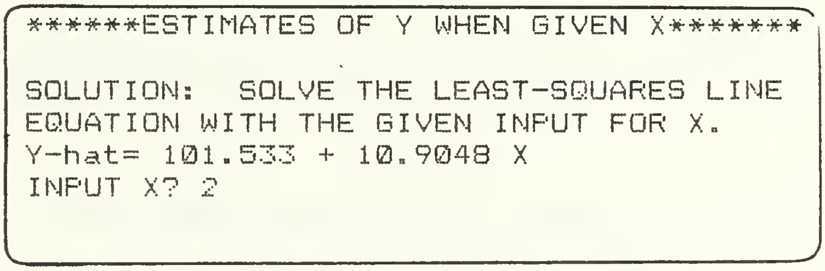


```
STEP 4.  THE FINAL STEP IS TO COMPUTE
THE INTERVAL USING THE PROPER FORMULA
(FORMULA 18).
THE PREDICTION LIMITS ARE:
LOWER= 101.594; UPPER= 145.091

SELECT: 1-MENU; 2-INPUT NEW DATA;
3-EXIT? 1
```

ESTIMATES OF Y WHEN GIVEN X EXAMPLE: Use the least squares line calculated from the master data set to predict the maintenance cost during the second year. (Ref 2, pg 438)



```
                          MENU
1-HYP. TEST
2-CONF. INT. (for alpha or beta)
3-CONF. INT. (mean of Y at Xo)
4-LIMITS OF PRED.
5-ESTIMATES OF y WHEN GIVEN x
6-EXIT
SELECTION? 5
```



```
******ESTIMATES OF Y WHEN GIVEN X*******

SOLUTION:  SOLVE THE LEAST-SQUARES LINE
EQUATION WITH THE GIVEN INPUT FOR X.
Y-hat= 101.533 + 10.9048 X
INPUT X? 2
```

At this point, the solution is calculated and briefly displayed on the screen before automatically returning to the main menu.

MODULE 8

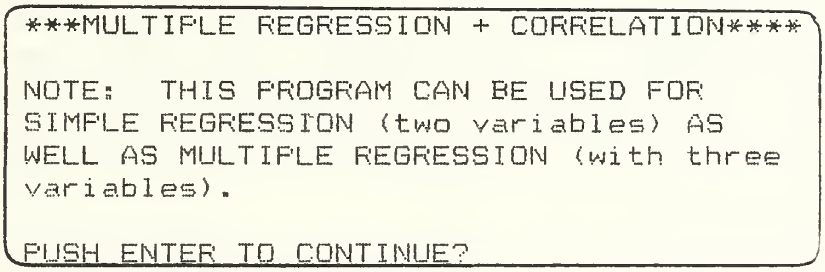
EXAMPLES AND SCREEN DISPLAYS

MASTER DATA SET FOR MODULE 8 EXAMPLES

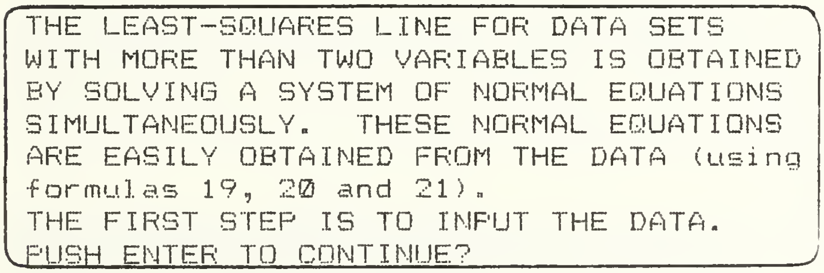
Area	Fines ($1,000)	Arrests (100's)	Number of Police Officers
1	76	15	25
2	38	10	29
3	25	7	15
4	98	21	25
5	93	14	11
6	54	8	13
7	78	14	22
8	85	24	14
9	65	9	13
10	88	23	11

(Ref 3, pg 180)

MULTIPLE REGRESSION EXAMPLE: Find the regression of fines on arrests and number of police officers.



```
***MULTIPLE REGRESSION + CORRELATION****

NOTE:  THIS PROGRAM CAN BE USED FOR
SIMPLE REGRESSION (two variables) AS
WELL AS MULTIPLE REGRESSION (with three
variables).

PUSH ENTER TO CONTINUE?
```



```
THE LEAST-SQUARES LINE FOR DATA SETS
WITH MORE THAN TWO VARIABLES IS OBTAINED
BY SOLVING A SYSTEM OF NORMAL EQUATIONS
SIMULTANEOUSLY.  THESE NORMAL EQUATIONS
ARE EASILY OBTAINED FROM THE DATA (using
formulas 19, 20 and 21).
THE FIRST STEP IS TO INPUT THE DATA.
PUSH ENTER TO CONTINUE?
```

```
INPUT THE NUMBER OF VARIABLES (2 or 3)?
3
```

```
INPUT THE NUMBER OF VARIABLES (2 or 3)?
3

INPUT THE NUMBER OF OBSERVATIONS? 10
```

```
1.   X1= ? 15
```

```
1.   X1= ? 15        X2= ? 25        Y= ?
```

At this point, the user continues to enter all the data triplets.

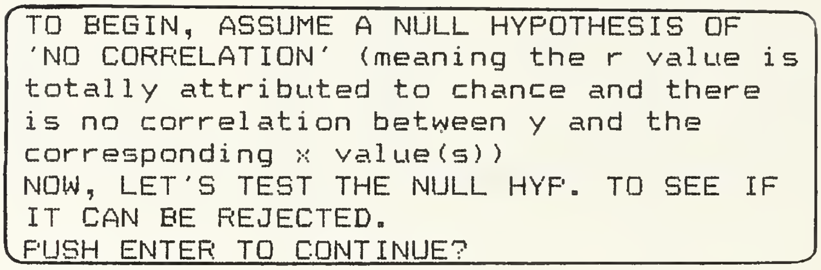
```
1.   X1= ? 15         X2= ? 25         Y= ? 76
2.   X1= ? 10         X2= ? 29         Y= ? 38
3.   X1= ? 7          X2= ? 15         Y= ? 25
4.   X1= ? 21         X2= ? 25         Y= ? 98
5.   X1= ? 14         X2= ? 11         Y= ? 93
6.   X1= ? 8          X2= ? 13         Y= ? 54
7.   X1= ? 14         X2= ? 22         Y= ? 78
```

```
8.   X1= ? 24         X2= ? 14         Y= ? 85
9.   X1= ? 9          X2= ? 13         Y= ? 65
10.  X1= ? 23         X2= ? 11         Y= ? 88

DATA INPUT IS COMPLETE!!
WOULD YOU LIKE TO REVIEW OR CHANGE ANY
OF YOUR INPUT DATA (Y or N)? N
```

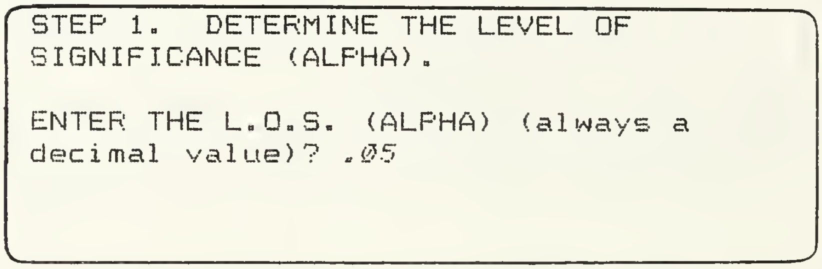
The user has the option to review and change any of the input data. Upon completion of these changes or if no changes are made, the computer will calculate and briefly display the NORMAL equation and the equation for the least-squares line before automatically branching to the MENU.

```
                     MENU
1-SIGNIFICANCE FOR R
2-COEFFICIENT OF DETERMINATION
3-ESTIMATES OF Y WHEN GIVEN X (X1, X2)
4-EXIT

SELECTION? 1
```

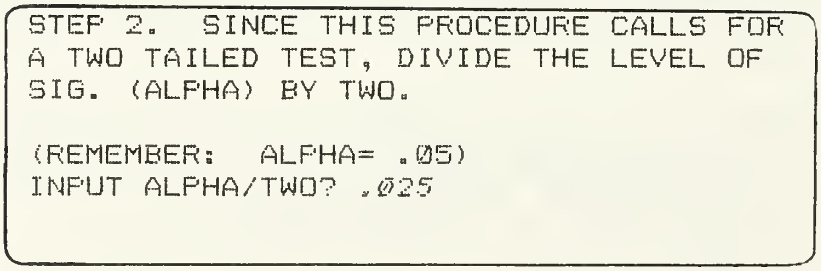
```
***********SIGNIFICANCE TEST************
*FOR THE COEFFICIENT OF CORRELATION(r)**
OBJECT:  TO DETERMINE IF 'r' MAY BE
ATTRIBUTED TO CHANCE AT A CERTAIN LEVEL
OF SIGNIFICANCE!

PUSH ENTER TO CONTINUE?
```



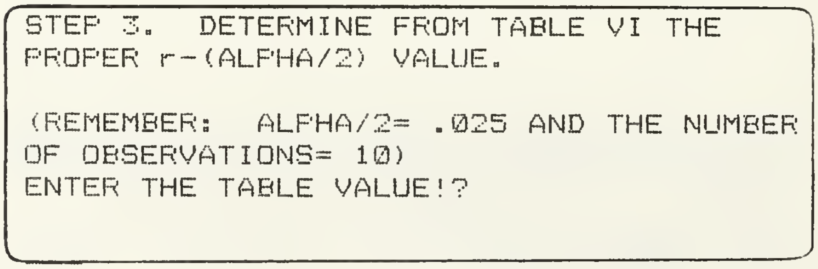
```
TO BEGIN, ASSUME A NULL HYPOTHESIS OF
'NO CORRELATION' (meaning the r value is
totally attributed to chance and there
is no correlation between y and the
corresponding x value(s))
NOW, LET'S TEST THE NULL HYP. TO SEE IF
IT CAN BE REJECTED.
PUSH ENTER TO CONTINUE?
```



```
STEP 1.  DETERMINE THE LEVEL OF
SIGNIFICANCE (ALPHA).

ENTER THE L.O.S. (ALPHA) (always a
decimal value)? .05
```

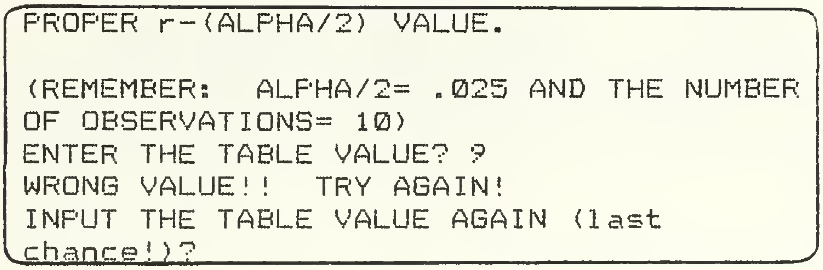


```
STEP 2.  SINCE THIS PROCEDURE CALLS FOR
A TWO TAILED TEST, DIVIDE THE LEVEL OF
SIG. (ALPHA) BY TWO.

(REMEMBER:  ALPHA= .05)
INPUT ALPHA/TWO? .025
```

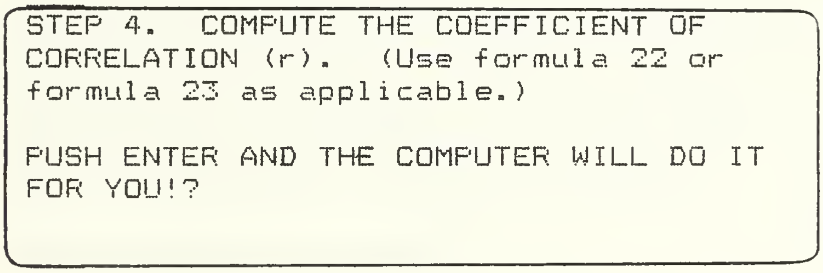


```
STEP 3.  DETERMINE FROM TABLE VI THE
PROPER r-(ALPHA/2) VALUE.

(REMEMBER:  ALPHA/2= .025 AND THE NUMBER
OF OBSERVATIONS= 10)
ENTER THE TABLE VALUE!?
```

The computer will verify user input at this point and provide error notification and correction if necessary.

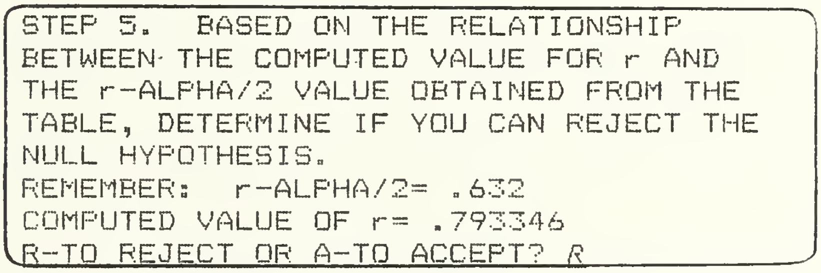


```
PROPER r-(ALPHA/2) VALUE.

(REMEMBER:  ALPHA/2= .025 AND THE NUMBER
OF OBSERVATIONS= 10)
ENTER THE TABLE VALUE? ?
WRONG VALUE!!  TRY AGAIN!
INPUT THE TABLE VALUE AGAIN (last
chance!)?
```



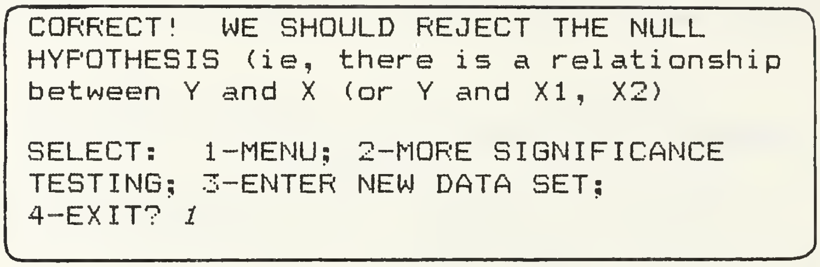
```
STEP 4.  COMPUTE THE COEFFICIENT OF
CORRELATION (r).  (Use formula 22 or
formula 23 as applicable.)

PUSH ENTER AND THE COMPUTER WILL DO IT
FOR YOU!?
```

The Coefficient of Correlation is computed and briefly displayed before branching to STEP 5.



```
STEP 5.  BASED ON THE RELATIONSHIP
BETWEEN THE COMPUTED VALUE FOR r AND
THE r-ALPHA/2 VALUE OBTAINED FROM THE
TABLE, DETERMINE IF YOU CAN REJECT THE
NULL HYPOTHESIS.
REMEMBER:  r-ALPHA/2= .632
COMPUTED VALUE OF r= .793346
R-TO REJECT OR A-TO ACCEPT? R
```

The computer will verify user input at this point and provide error notification and correction if necessary.



```
CORRECT!  WE SHOULD REJECT THE NULL
HYPOTHESIS (ie, there is a relationship
between Y and X (or Y and X1, X2)

SELECT:  1-MENU; 2-MORE SIGNIFICANCE
TESTING; 3-ENTER NEW DATA SET;
4-EXIT? 1
```

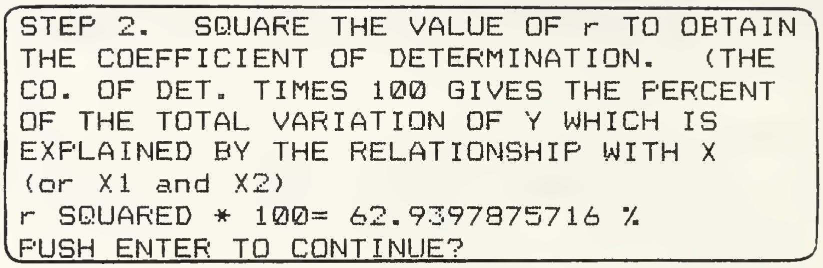
COEFFICIENT OF DETERMINATION EXAMPLE: Using the master data set, find and interpret the Coefficient of Determination.

```
                   MENU
1-SIGNIFICANCE FOR R
2-COEFFICIENT OF DETERMINATION
3-ESTIMATES OF Y WHEN GIVEN X (X1, X2)
4-EXIT

SELECTION? 2
```

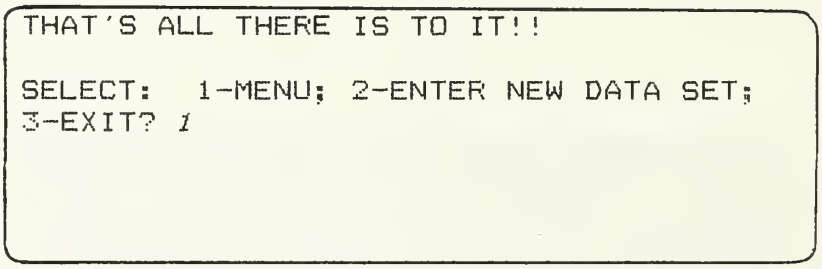
```
******COEFFICIENT OF DETERMINATION******
OBJECT:  DETERMINE HOW MUCH OF THE TOTAL
VARIATION AMONG THE Y VALUES IN A DATA
SET CAN BE ATTRIBUTED TO A RELATIONSHIP
WITH THE OTHER VARIABLES IN THE DATA
SET AND HOW MUCH IS DUE TO CHANCE.

PUSH ENTER TO CONTINUE?
```

```
STEP 1.  CALCULATE THE COEFFICIENT OF
CORRELATION (r).  (Use formula 22 or
formula 23 as applicable.)

PUSH ENTER AND THE COMPUTER WILL ASSIST
YOU?
```

The Coefficient of Correlation is computed and briefly displayed before branching to STEP 2.

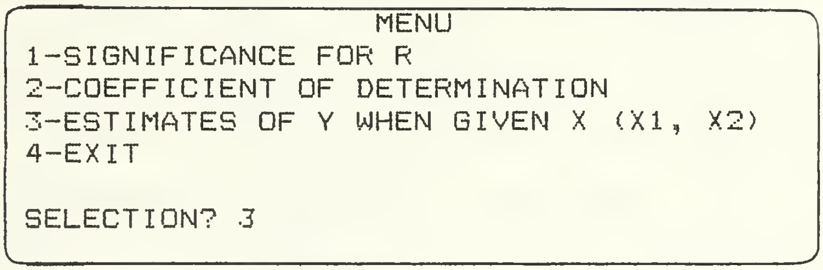


```
STEP 2.  SQUARE THE VALUE OF r TO OBTAIN
THE COEFFICIENT OF DETERMINATION.  (THE
CO. OF DET. TIMES 100 GIVES THE PERCENT
OF THE TOTAL VARIATION OF Y WHICH IS
EXPLAINED BY THE RELATIONSHIP WITH X
(or X1 and X2)
r SQUARED * 100= 62.9397875716 %
PUSH ENTER TO CONTINUE?
```

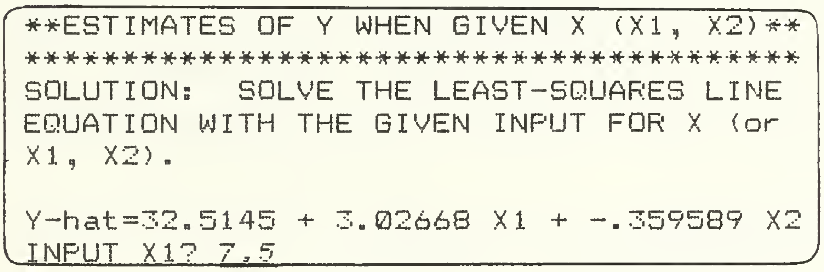


```
THAT'S ALL THERE IS TO IT!!

SELECT:  1-MENU; 2-ENTER NEW DATA SET;
3-EXIT? 1
```

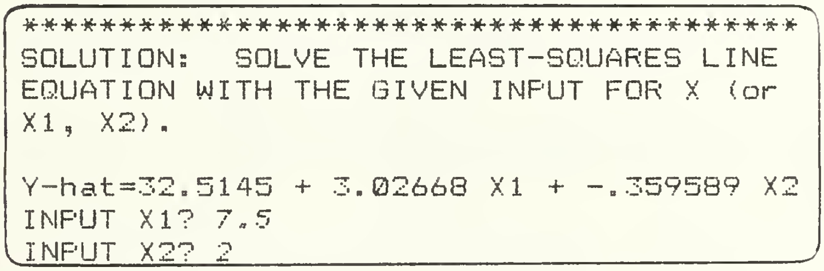
ESTIMATES OF Y WHEN GIVEN X1 AND X2 EXAMPLE: Predict the amount of fines given 750 arrests and 5 police officers.



```
                    MENU
1-SIGNIFICANCE FOR R
2-COEFFICIENT OF DETERMINATION
3-ESTIMATES OF Y WHEN GIVEN X (X1, X2)
4-EXIT

SELECTION? 3
```



```
**ESTIMATES OF Y WHEN GIVEN X (X1, X2)**
****************************************
SOLUTION:  SOLVE THE LEAST-SQUARES LINE
EQUATION WITH THE GIVEN INPUT FOR X (or
X1, X2).

Y-hat=32.5145 + 3.02668 X1 + -.359589 X2
INPUT X1? 7.5
```



```
****************************************
SOLUTION:  SOLVE THE LEAST-SQUARES LINE
EQUATION WITH THE GIVEN INPUT FOR X (or
X1, X2).

Y-hat=32.5145 + 3.02668 X1 + -.359589 X2
INPUT X1? 7.5
INPUT X2? 2
```

At this point, the solution is calculated and displayed briefly before the program automatically branches back to the main menu.

STANDARD FORMULAS AND TABLE DESCRIPTIONS

This section contains a listing of some standard statistical formulas to aid the student in using the program modules. In addition, a brief description of the statistical tables used by the modules is provided. The number adjacent to each formula is a key value which is referenced by the modules. Similarly, a roman numeral uniquely identifies each table description. The actual tables referenced by the program modules can be found in reference 2.

Formulas

1. $nPr=\frac{n!}{(n-r)!}$

2. $\binom{n}{r}=\frac{n!}{r!(n-r)!}$

3. $P(B_i/A)=\frac{P(B_i)\cdot P(A/B_i)}{P(B_1)\cdot P(A/B_1)+P(B_2)\cdot P(A/B_2)+\cdots+P(B_k)\cdot P(A/B_k)}$

4. $f(x)=\binom{n}{x}p^x(1-p)^{n-x}$ for $x=0,1,2...,$ or n

5. $f(k)=p(1-p)^{(k-1)}$

6. $P(k)=1-(1-p)^k$

7. $Q(k)=(1-p)^k$

8. $f(x)=\frac{(np)^x\cdot e^{-np}}{x!}$ $x=0,1,2...$

9. prob. value between o and x for exponential distribution$=1-e^{-x/u}$

10. $\left[n = \frac{z_{\alpha/2}\, s}{E}\right]^2$

11. $\bar{x} - t_{\alpha/2} \cdot \frac{s}{\sqrt{n}} < \mu < \bar{x} + t_{\alpha/2} \cdot \frac{s}{\sqrt{n}}$

12. $\frac{s}{1+\frac{z_{\alpha/2}}{\sqrt{2n}}} < \sigma < \frac{s}{1-\frac{z_{\alpha/2}}{\sqrt{2n}}}$

13. $\chi^2 = \frac{(n-1)s^2}{\sigma_0^2}$

14. $\sqrt{\frac{(n-1)s^2}{\chi^2_{\alpha/2}}} < \sigma < \sqrt{\frac{(n-1)s^2}{\chi^2_{1-\alpha/2}}}$

15. $t = \frac{a-\alpha}{Se\sqrt{\frac{1}{n}+\frac{n\cdot\bar{x}^2}{n(\Sigma x^2)-(\Sigma x)^2}}}$ or $\frac{b-\beta}{Se}\sqrt{\frac{n(\Sigma x^2)-(\Sigma x)^2}{n}}$

16. $a \pm t_{\alpha/2} \cdot Se\sqrt{\frac{1}{n}+\frac{n\cdot\bar{x}^2}{n(\Sigma x^2)-(\Sigma x)^2}}$ or $b \pm \frac{t_{\alpha/2} \cdot Se}{\sqrt{\frac{n(\Sigma x^2)-(\Sigma x)^2}{n}}}$

17. $(a+bx_0) \pm t_{\alpha/2} \cdot Se\sqrt{\frac{1}{n}+\frac{n(x_0-\bar{x})^2}{n(\Sigma x^2)-(\Sigma x)^2}}$

18. $(a+bx_0) \pm t_{\alpha/2} \cdot Se\sqrt{1+\frac{1}{n}+\frac{n(x_0-\bar{x})^2}{n(\Sigma x^2)-(\Sigma x)^2}}$

19. $\Sigma y = n \cdot b_0 + b_1(\Sigma x_1) + b_2(\Sigma x_2)$

20. $\Sigma x_1 y = b_0(\Sigma x_1) + b_1(\Sigma x_1^2) + b_2(\Sigma x_1 x_2)$

21. $\Sigma x_2 y = b_0(\Sigma x_2) + b_1(\Sigma x_1 x_2) + b_2(\Sigma x_2^2)$

22. $$r = \frac{n(\Sigma xy)-(\Sigma x)(\Sigma y)}{\sqrt{n(\Sigma x^2)-(\Sigma x)^2}\ \sqrt{n(\Sigma y^2)-(\Sigma y)^2}}$$

23. $$r = \sqrt{1-\frac{\Sigma(y-\hat{y})2}{\Sigma(y-\bar{y})^2}}$$

Table Descriptions

Table I: Table I contains the probabilities that a random variable having the standard normal distribution will take on a value between 0 and z.

Table II: Table II displays the values for the area under the t-distribution curve to the right of t-alpha.

Table III: Table III displays the values for the area under the chi-square distribution curve to the right of chi-square-alpha.

Table V: Table V is used to obtain binomial probabilities given n, x, and p.

Table VI: Table VI displays the values of r for n between 3 and 92 and for $r_{.025}$, $r_{.010}$ and $r_{.005}$.

Table IX: Table IX is used to obtain binomial coefficients for n between 0 and 20 and x between 0 and 10.

APPENDIX B

This section contains the program listings for the developed CAI package. Modules 1 through 8 are listed in sequential order.

```
0 'TITLE: MODULE 1
2 'DATE: 15 MAY 85
4 'PURPOSE: THIS PROGRAM IS DESIGNED TO ASSIST THE STUDENT
IN THE CALCULATION OF PERMUTATIONS, COMBINATIONS, AND IN
SOLVING PROBLEMS INVOLVING BAYES' THEOREM.
6 'A SEPARATE SUBROUTINE IS DEFINED FOR EACH OF THE ABOVE
APPLICATIONS.
8 '
10 'INPUT VARIABLES:
12 'EVENTS=NUMBER OF EVENTS
14 'N=TOTAL NUMBER OF DISTINCT OBJECTS
16 'PA=ARRAY OF THE PROBABILITIES-P(B)
18 'PB=ARRAY OF THE PROBABILITIES-P(A/B)
20 '
22 'OUTPUT VARIABLES
24 'BAY=ARRAY OF POSTERIOR PROBABILITIES
26 'BTTM=PROBABILITY P(A)
28 'CMB=NUMBER OF COMBINATIONS
30 'X=NUMBER OF PERMUTATIONS
32 '
34 'INTERNAL VARIABLES
36 'I/J=LOOP INDICIES
38 'I1DEX/I2DEX/I3DEX = SCREEN FORMAT VARIABLES
40 'J/NSUB=SUBSTITUTION VARIABLES FOR N (number of distinct
objects)
42 'Q=FLOW CONTROL
44 'SUM=SUM OF THE PROBABILITIES P(B)
46 'TP=NUMERATOR IN THE BAYES' CALCULATION
48 '
480 '***************************************
485 'TOTAL PROGRAM MENU SELECTION
490 '***************************************
500 CLS:PRINTTAB(18);"MENU"
510 PRINT"1-PERMUTATIONS"
520 PRINT"2-COMBINATIONS"
530 PRINT"3-BAYES' THEOREM"
540 PRINT"4-EXIT":PRINT
550 INPUT "SELECTION";Q
560 ON Q GOTO 1000,2000,3000,1110
```

```
570 END
970 '========================================
975 '========================================
980 '****************************************
985 'PERMUTATION DRIVER AND OUTPUT ROUTINE
990 '****************************************
1000 CLS:PRINT"*************PERMUTATIONS*************"
1010 PRINT"THIS PROGRAM CALCULATES THE NUMBER OF
PERMUTATIONS OF r OBJECTS SELECTED FROM A SET OF n DISTINCT
OBJECTS":PRINT
1020 CLEAR:ON ERROR GOTO 22000
1030 GOSUB 1400
1060 GOSUB 1500
1070 GOSUB 10000
1080 CLS:PRINT"THE NUMBER OF PERMUTATIONS OF";R;"OBJECTS
SELECTED FROM A SET OF";N;"DISTINCT       OBJECTS IS
=";X:PRINT
1090 INPUT "SELECT: 1-MENU; 2-MORE PERMUTATIONS;
3-EXIT";Q
1100 ON Q GOTO 500,1000,1110
1110 MENU
1380 '****************************************
1385 'COMMON INPUT ROUTINE (also used in the Combinations
subroutine)
1390 '****************************************
1400 INPUT "INPUT n";N
1410 IF N<>ABS(INT(N)) THEN PRINT"n MUST BE A POSITIVE
INTEGER!":GOTO 1400
1420 INPUT "INPUT r";R
1430 IF R<>ABS(INT(R)) OR R>N THEN PRINT"r MUST BE A
POSITIVE INTEGER <= n.":GOTO 1420
1440 RETURN
1480 '****************************************
1485 'DISPLAY OF THE NECESSARY COMPUTATIONS
1490 '****************************************
1500 CLS:PRINT"USE THE FOLLOWING FORMULA TO CALCULATE  THE
ANSWER. (Formula-1)"
1510 PRINT"nPr=n!/(n-r)!=";N;"!/(";N;"-";R;")!"
1515 IF R=0 THEN RETURN
1520 PRINT"THE CALCULATION CAN BE SIMPLIFIED TO:"
1530 FOR I=N TO (N-R)+1 STEP -1
1540  PRINTI;
1550  IF I>(N-R)+1 THEN PRINT"*";
1560 NEXT I
1570 PRINT:INPUT "PUSH ENTER FOR A COMPUTER SOLUTION!";Q
1580 RETURN
1970 '========================================
1975 '========================================
1980 '****************************************
1985 'COMBINATIONS DRIVER AND OUTPUT ROUTINE
```

```
1990 '*****************************************
2000 CLS:PRINT"***************COMBINATIONS**************"
2005 CLEAR:ON ERROR GOTO 22000
2010 PRINT"THIS PROGRAM CALCULATES THE NUMBER OF    WAYS IN
WHICH r OBJECTS CAN BE SELECTED FROM A SET OF n DISTINCT
OBJECTS.":PRINT
2020 GOSUB 1400
2030 GOSUB 2500
2040 GOSUB 20000
2050 CLS:PRINT"THE NUMBER OF WAYS IN WHICH ";RSUB;"OBJECTS
CAN BE SELECTED FROM A SET OF";NSUB;"DISTINCT  OBJECTS IS
=";CMB:PRINT
2060 INPUT "SELECT: 1-MENU; 2-MORE COMBINATIONS;
3-EXIT";Q
2070 ON Q GOTO 500,2000,1110
2080 END
2480 '*****************************************
2485 'DISPLAY OF THE NECESSARY CALCULATIONS
2490 '*****************************************
2500 CLS:PRINT"USE THE FOLLOWING FORMULA TO CALCULATE  THE
ANSWER (Formula-2) OR OBTAIN THE     ANSWER DIRECTLY FROM
TABLE IX."
2510 PRINT"ANSWER = n!/r!(n-r)!
=";N;"!/(";R;"!)*(";(N-R);"!)"
2520 IF R=0 THEN RETURN
2530 PRINT"THE CALCULATION CAN BE SIMPLIFIED TO:"
2535 RSUB=R
2540 PRINT"(";
2545 IF N-R<R THEN R=N-R
2550 FOR I=N TO (N-R)+1 STEP -1
2560  PRINT I;
2570  IF I>(N-R)+1 THEN PRINT"*";
2580 NEXT I
2590 PRINT") /";R;"!"
2600 INPUT "PUSH ENTER FOR A COMPUTER SOLUTION!";Q
2610 RETURN
2970 '=========================================
2975 '=========================================
2980 '*****************************************
2985 'BAYES' DRIVER ROUTINE
2990 '*****************************************
3000 CLS:PRINT"**************BAYES' THEOREM*************"
3020 PRINT"METHOD: FORMAT THE GIVEN DATA INTO A     BAYES'
TABLE, THEN CALCULATE THE        POSTERIOR PROBABILITIES
USING FORMULA-3":PRINT
3030 INPUT "PUSH ENTER TO CONTINUE";Q
3040 CLEAR:ON ERROR GOTO 22000
3045 GOSUB 3100
3050 GOSUB 3200
3060 GOSUB 15000
```

```
3070 GOSUB 3400
3080 INPUT "SELECT: 1-MENU; 2-MORE BAYES; 3-EXIT";Q
3090 ON Q GOTO 500,3040,1110
3095 END
3097 '****************************************
3098 'TABLE FORMATING FOR DATA INPUT
3099 '****************************************
3100 CLS:PRINT"STEP 1. DETERMINE THE NUMBER OF DISTINCT
EVENTS (B).":PRINT
3110 INPUT "INPUT NUMBER OF EVENTS (B)";EVENTS
3120 IF EVENTS>5 THEN PRINT"SORRY THIS PROGRAM EXECUTES A
MAXIMUM OF 5 EVENTS!":END
3130 IF ABS(INT(EVENTS))<>EVENTS THEN PRINT"TRY AGAIN":GOTO
3110
3140 CLS:PRINT"STEP 2. BUILD THE BAYES' TABLE: FIRST
DETERMINE THE CAUSE PROBABILITIES THEN  DETERMINE THE
CONDITIONAL PRIOR          PROBABILITIES. (These
probabilities are usually given in the problem or are
easily calculated.)":PRINT
3150 INPUT "PUSH ENTER TO CONTINUE";Q
3160 RETURN
3200 CLS:PRINT"EVENT"
3210 FOR I=1 TO EVENTS
3220  PRINTTAB(1);"B";I
3230 NEXT I
3240 PRINT@6,"CAUSE PROB:PRIOR PROB :BAYES PROB";
3250 I1DEX=46:I2DEX=57:I3DEX=69:SUM=0
3260 FOR I=1 TO EVENTS
3270  PRINT@240,"INPUT P(B";I;")   ";:INPUT PB(I)
3280  PRINT@255,"          ";
3290  IF PB(I)>=1 OR PB(I)<0 THEN GOTO 3270
3300  PRINT@I1DEX,"P(B)=";PB(I);
3310  PRINT@240,"INPUT P(A/B";I;")";:INPUT PA(I)
3320  PRINT@255,"       ";
3330  IF PA(I)>=1 OR PA(I)<0 THEN GOTO 3310
3340  PRINT@I2DEX,"P(A/B)=";PA(I);
3350  I1DEX=I1DEX+40:I2DEX=I2DEX+40
3360  SUM=PB(I)+SUM
3370 NEXT I
3380 IF SUM>1.0001 OR SUM<.99 THEN PRINT"THE SUM OF THE
CAUSE PROBABILITIES DO   NOT = 1!! TRY AGAIN.":FOR I=1 TO
1000:NEXT I:GOTO 3200
3390 PRINT@240,"STEP 4. PUSH ENTER FOR BAYES' PROB.";:INPUT
Q
3395 RETURN
3400 FOR I=1 TO EVENTS
3410  PRINT@I3DEX,"P(B/A)=";INT(BAY(I)*100+.5)/100
3420  I3DEX=I3DEX+40
3430 NEXT I
3440 PRINT@240,"P(A)=";BTTM;"                              "
```

```
3450 RETURN
3980 '****************************************
3985 'OUTPUT ROUTINE
3990 '****************************************
9970 '========================================
9975 '========================================
9980 '****************************************
9985 'CALCULATION OF PERMUTATIONS
9990 '****************************************
10000 IF R=0 THEN X=1:RETURN
10010 IF R=1 THEN X=N:RETURN
10020 X=N
10030 FOR I=(N-1) TO (N-R)+1 STEP -1
10040  X=X*I
10050 NEXT I
10060 RETURN
14980 '****************************************
14985 'BAYES' CALCULATIONS
14990 '****************************************
15000 FOR I=1 TO EVENTS
15010  BTTM=BTTM+PB(I)*PA(I)
15020 NEXT I
15030 FOR J=1 TO EVENTS
15040  TP=PB(J)*PA(J)
15060  BAY(J)=TP/BTTM
15070 NEXT J
15080 RETURN
19980 '****************************************
19985 'CALCULATION OF COMBINATIONS
19990 '****************************************
20000 J=N:NSUB=N
20020 IF R=0 THEN J=1:GOTO 20090
20030 IF R=1 THEN GOTO 20090
20040 FOR I=1 TO R-1
20050  N=N-1
20060  J=J*N
20070 NEXT I
20090 N=R
20100 GOSUB 10000
20110 CMB=J/X
20120 RETURN
21980 '****************************************
21985 'ERROR TRAP
21990 '****************************************
22000 IF ERR=6 THEN PRINT"RESULTING NUMBER EXCEEDS COMPUTER
    CAPACITY.":FOR I=1 TO 1600:NEXT I:GOTO 500
22020 PRINT"ERROR";ERR;"IN LINE";ERL
22030 PRINT"CONSULT OPERATORS MANUAL":END
```

```
0 'TITLE: MODULE 2
2 'DATE: 15 MAY 85
4 'PURPOSE: THIS PROGRAM IS DESIGNED AS AN AID IN GENERATING
RANDOM SAMPLES OF VARIOUS TYPES. SUMMARY STATISTICS ARE KEPT
FOR EACH OF THE GENERATED SAMPLES.
6 'THIS MODULE DIFFERS FROM THE PREVIOUS AND SUBSEQUENT
MODULES IN THAT IT IS DESIGNED AS AN APPLICATIONS MODULE
ONLY AND DOES NOT ATTEMPT TO PROVIDE THE STUDENT WITH A
FRAMEWORK TO PRODUCE RANDOM NUMBERS.
8 '
10 'SUBROUTINES INCLUDE:
12 'RANDOM SAMPLES DRAWN FORM A FINITE POPULATION
14 'EXPONENTIALLY DISTRIBUTED RANDOM SAMPLES
16 'NORMALLY DISTRIBUTED RANDOM SAMPLES
18 'UNIFORMLY DISTRIBUTED RANDOM SAMPLES
20 '
22 'INPUT VARIABLES
24 'A1=PROBABILITY VALUE
26 'LNUM=NUMBER OF RANDOM NUMBERS NEEDED
28 'OPAIRS=NUMBER OF POSSIBLE OUTCOMES
30 'PROB=INDIVIDUAL PROBABILITIES
32 'SEED=SEED VALUE TO START RANDOM NUMBER GENERATION
34 '
36 'OUTPUT VARIABLES
38 'ASTDV=STANDARD DEVIATION (normal dist.)
40 'CAT=CORRESPONDING OUTCOME
42 'MEAN=MEAN OF THE RANDOM NUMBER SET
44 'RAD=RANDOM NUMBER
46 'R1AD=RANDOM NUMBER
48 'STDV=STANDARD DEVIATION
50 '
52 'INTERNAL VARIABLES
54 'A2=RANDOM LABLE
56 'I/J/K=LOOP INDICIES
58 'LIM=INDIVIDUAL RANDOM LABEL UPPER LIMIT
60 'MARK=SCREEN FORMAT VARIABLE
62 'Q1/Q2/Q3=SCREEN FORMAT VARIABLES
64 'SCAT=SUM OF THE CORRESPONDING OUTCOMES
66 'S2CAT=SUM OF THE SQUARED CORRESPONDING OUTCOMES
68 'U1/U2=RANDOM NUMBERS (used in generation of the normally
dist. ran. num.)
70 'XSUM=SUM OF THE EXP. DIST. RANDOM NUMBERS
72 'X2SUB=SUM OF THE SQUARED EXP. DIST. RANDOM NUMBERS
74 'Y=USED TO INITIALIZE THE RANDOM NUMBER SEQUENCE
76 '
480 '*************************************
485 'TOTAL PROGRAM MENU SELECTION
490 '*************************************
500 CLS:PRINTTAB(18);"MENU"
510 PRINT"1-RANDOM SAMPLE-FINITE POP."
```

```
520 PRINT"2-RANDOM SAMPLE-EXPONENTIAL"
530 PRINT"3-RANDOM SAMPLE-NORMAL"
540 PRINT"4-RANDOM SAMPLE-UNIFORM"
550 PRINT"5-EXIT":PRINT
560 INPUT"SELECTION";Q
570 ON Q GOTO 700,3000,4000,5000,600
580 END
600 MENU
670 '======================================
675 '======================================
680 '**************************************
685 'INTRODUCTION AND DRIVER ROUTINE FOR OBTAINING A RANDOM
SAMPLE FROM A FINITE POPULATION
690 '**************************************
700 CLS:PRINT"******RANDOM SAMPLE (Finite Pop.)*******"
710 PRINT"THIS PROGRAM IS DESIGNED TO PRODUCE     SUMMARY
STATISTICS FROM A RANDOM SAMPLE. THE RANDOM SAMPLE IS BASED
ON          PROBABILITIES INPUT BY THE USER.":PRINT
720 INPUT "PUSH ENTER TO CONTINUE";Q
730 CLS:PRINT"METHOD: A TABLE CONTAINING THE 'POSSIBLE
OUTCOMES'; 'PROBABILITY'OF EACH OUTCOME; AND THE RANGE OF
'RANDOM LABELS' IS   PRODUCED BY THE USER (with computer
aid). ";
740 PRINT"THE COMPUTER THEN GENERATES THE RANDOM NUMBERS AND
SELECTS THE PROPER OUTCOME FOR EACH EVENT BASED ON   THE
VALUES IN THE TABLE! ";
750 INPUT "PUSH ENTER TO CONT.";Q
760 CLS:PRINT"NOTE: ONLY INDIVIDUAL     PROBABILITIES ARE
NEEDED AS THE COMPUTER GENERATES THE CUMULATIVE
PROBABILITIES."
770 INPUT "TO BEGIN TABLE GENERATION PUSH ENTER";Q
780 CLEAR:ON ERROR GOTO 22000
785 GOSUB 1000
790 GOSUB 2000
795 FOR I=1 TO 500:NEXT I
800 INPUT "SELECT: 1-MENU; 2-MORE RANDOM SAMPLES;  3-EXIT";Q
810 ON Q GOTO 500,780,600
880 '**************************************
885 'DATA INPUT (TABLE FORMAT!)
890 '**************************************
1000 CLS:DEFSNG A-Z
1005 INPUT"HOW MANY OUTCOMES (for exp: there are 4 possible
outcomes in the example on page 295)";OPAIRS
1010 IF OPAIRS<2 OR ABS(INT(OPAIRS))<>OPAIRS THEN GOTO 1000
1020 DIM A1(OPAIRS):DIM A2(OPAIRS)
1030 CLS:Q1=40:Q2=54:Q3=66
1040 FOR I=1 TO OPAIRS
1050  IF Q1=40 THEN PRINT
TAB(3);"OUTCOME";TAB(17);"PROB.";TAB(27);"RANDOM LBL."
1060  PRINT @Q1,I;".";
```

```
1070  INPUT A1(I)
1080  PRINT @Q2," ";
1090  INPUT PROB
1095  IF PROB>1 OR PROB<0 THEN PRINT:PRINT"PROB MUST BE IN
POSITIVE DECIMAL FORM!":GOTO 1080
1100  PRINT @Q3,LIM*1000;"-";((LIM+PROB)*1000)-1
1110  LIM=LIM+PROB
1115  A2(I)=LIM*1000
1120  Q1=Q1+40:Q2=Q2+40:Q3=Q3+40
1130  IF Q1>240 THEN CLS:Q1=40:Q2=54:Q3=66
1140  NEXT I
1150 PRINT"DATA TABLE INPUT IS COMPLETE!"
1160 RETURN
1980 '****************************************
1985 'RANDOM NUMBER GENERATION AND SUMMARY STATISTICS
1990 '****************************************
2000 CAT=0:SCAT=0:S2CAT=0:STDV=0:MEAN=0
2005 INPUT "HOW MANY RANDOM NUMBERS DO YOU NEED";UPLM
2010 INPUT "SEED";SEED
2015 SEED=INT(ABS(SEED))
2020 FOR I=0 TO SEED
2030  Y=RND(1)
2040 NEXT I
2045 Q1=45:Q2=70:CLS
2050 FOR I=1 TO UPLM
2060  RAD=INT(RND(1)*A2(OPAIRS))
2070  IF Q1=45 THEN PRINT"RANDOM NO.";TAB(18);"CORRESPONDING
OUTCOME"
2080  GOSUB 2500
2090  PRINTI".";
2100  PRINT@Q1,RAD;
2110  PRINT@Q2,CAT
2120  SCAT=SCAT+CAT:S2CAT=S2CAT+CAT^2
2125  Q1=Q1+40:Q2=Q2+40
2130  FOR K=1 TO 320:NEXT K
2140  IF Q1>270 THEN CLS:Q1=45:Q2=70
2150  NEXT I
2160 MEAN=SCAT/UPLM
2170 STDV=SQR((UPLM*S2CAT-SCAT^2)/(UPLM*(UPLM-1)))
2180 PRINT"SUMMARY STATISTICS FOR THIS SAMPLE:"
2190 PRINT"SAMPLE MEAN=";MEAN
2200 PRINT"STD. DEV.(s)=";STDV
2210 RETURN
2480 '****************************************
2485 'SELECTION OF OUTCOME
2490 '****************************************
2500 FOR J=1 TO OPAIRS
2510   IF RAD<A2(J) THEN CAT=A1(J):RETURN
2520 NEXT J
2530 END
```

```
2970 '========================================
2975 '========================================
2980 '****************************************
2985 'DATA INPUT/OUTPUT ROUTINE FOR AN EXPONENTIAL RANDOM
SAMPLE
2990 '****************************************
3000 CLS:PRINT"*******RANDOM SAMPLE - EXPONENTIAL******"
3010 PRINT"THIS PROGRAM IS DESIGNED TO GENERATE A  SERIES OF
EXPONENTIALLY DISTRIBUTED     RANDOM NUMBERS AND COMPUTES
THE         ASSOCIATED SUMMARY STATISTICS!":PRINT
3020 INPUT "PUSH ENTER TO CONTINUE";Q
3030 CLS:CLEAR:DEFSNG A-Z:ON ERROR GOTO 22000
3035 INPUT "INPUT MU (mean)";MEAN
3040 INPUT "HOW MANY RANDOM NUMBERS DO YOU NEED";LNUM
3045 IF LNUM<2 OR ABS(INT(LNUM))<>LNUM THEN PRINT"VALUE MUST
BE A POSITIVE INTEGER >= 2":GOTO 3040
3050 INPUT "INPUT SEED";SEED
3055 SEED=ABS(INT(SEED))
3060 GOSUB 3100
3070 PRINT"SUMMARY STATISTICS FOR THIS SAMPLE:"
3080 PRINT"MEAN=";MEAN
3085 PRINT"STD.DEV (s)=";STDV:PRINT
3090 INPUT "SELECT: 1-MENU; 2-MORE EXPONENTIAL      SAMPLES;
3-EXIT";Q
3095 ON Q GOTO 500,3030,600
3096 END
3097 '****************************************
3098 'RANDOM NUMBER GENERATION AND SUMMARY STATISTICS
3099 '****************************************
3100 FOR I=1 TO SEED
3110  Y=RND(1)
3120 NEXT I
3130 CLS:PRINTTAB(14);"RANDOM NUMBER"
3140 FOR I=1 TO LNUM
3145  MARK=MARK+1
3150  IF MARK=7 THEN CLS:MARK=1:PRINTTAB(14);"RANDOM NUMBER"
3160  RAD=RND(1)
3170  R1AD=INT(-MEAN*LOG(RAD))
3180  XSUM=XSUM+R1AD:X2SUM=X2SUM+R1AD^2
3190  PRINTTAB(14);I;".";R1ADR
3200  FOR K=1 TO 150:NEXT K
3210 NEXT I
3220 PRINT"GENERATION COMPLETE!"
3230
MEAN=XSUM/LNUM:STDV=SQR((LNUM*X2SUM-XSUM^2)/(LNUM*(LNUM-1)))
3240 RETURN
3970 '========================================
3975 '========================================
3980 '****************************************
3985 'DATA INPUT/OUTPUT FOR A NORMALLY DISTRIBUTED RANDOM
```

```
SAMPLE
3990 '****************************************
4000 CLS:PRINT"*********RANDOM SAMPLE - NORMAL*********"
4010 PRINT"THIS PROGRAM IS DESIGNED TO GENERATE A  SERIES OF
NORMALLY DISTRIBUTED RANDOM   NUMBERS AND TO COMPUTE THE
ASSOCIATED   SUMMARY STATISTICS!":PRINT
4020 INPUT "PUSH ENTER TO CONTINUE";Q
4030 CLS:CLEAR:DEFSNG A-Z:ON ERROR GOTO 22000
4035 INPUT "INPUT THE MU (mean)";MEAN
4040 INPUT "INPUT THE STAND. DEV.";STDV
4050 INPUT "HOW MANY RANDOM SAMPLES DO YOU NEED";LNUM
4055 IF LNUM<2 OR ABS(INT(LNUM))<>LNUM THEN GOTO 4050
4060 INPUT "INPUT SEED(for random sample)";SEED
4070 SEED=INT(ABS(SEED))
4090 GOSUB 4200
4100 CLS:PRINT"SUMMARY STATISTICS FOR THIS SAMPLE:"
4110 PRINT"SAMPLE MEAN(Xbar)=";XMEAN
4120 PRINT"SAMPLE STDV(s)=";ASTDV:PRINT
4130 INPUT "SELECT: 1-MENU; 2-MORE NORMALLY DIST.   SAMPLES;
3-EXIT";Q
4140 ON Q GOTO 500,4030,600
4150 END
4180 '****************************************
4185 'RANDOM NUMBER GENERATION AND SUMMARY STATISTICS
4190 '****************************************
4200 FOR I=1 TO SEED
4210  Y=RND(1)
4220 NEXT I
4230 CLS:PRINTTAB(14);"RANDOM NUMBER"
4240 FOR I=1 TO LNUM
4250  MARK=MARK+1
4260  IF MARK>6 THEN CLS:MARK=1:PRINTTAB(14);"RANDOM NUMBER"
4270  U1=RND(1):U2=RND(1)
4280  RAD=SQR(-2*LOG(U1))*COS(2*3.141592654*U2)*STDV+MEAN
4290  XSUM=XSUM+RAD
4300  X2SUM=X2SUM+RAD^2
4310  PRINTTAB(12);I;":";RAD
4320  FOR K=1 TO 130:NEXT K
4330 NEXT I
4340 PRINT"RANDOM NUMBER GENERATION COMPLETE!"
4350
XMEA
=XSUM/LNUM:ASTDV=SQR((LNUM*X2SUM-XSUM^2)/(LNUM*(LNUM-1)))
4360 FOR I=1 TO 500:NEXT I
4370 RETURN
4970 '========================================
4975 '========================================
4980 '****************************************
4985 'DATA INPUT/OUTPUT FOR A UNIFORMLY DISTRIBUTED RANDOM
SAMPLE
```

```
4990 '****************************************
5000 CLS:PRINT"*********RANDOM SAMPLE - UNIFORM*********"
5010 PRINT"THIS PROGRAM IS DESIGNED TO GENERATE A  SERIES OF
UNIFORMLY DISTRIBUTED RANDOM  NUMBERS AND TO COMPUTE THE
ASSOCIATED   SUMMARY STATISTICS.!":PRINT
5020 INPUT "PUSH ENTER TO CONTINUE";Q
5030 CLS:CLEAR:DEFSNG A-Z:ON ERROR GOTO 22000
5040 INPUT "INPUT THE DESIRED UPPER LIMIT(>=0)";UPLM
5050 INPUT "INPUT THE DESIRED LOWER LIMIT(>=0)";LLM
5055 IF UPLM<LLM OR UPLM*LLM<0 THEN PRINT"YOUR LIMIT VALUES
ARE IN THE WRONG ORDER OR ARE NEGATIVE: TRY AGAIN!":GOTO
5040
5060 INPUT "HOW MANY RANDOM LABELS(Numbers) DO YOU
NEED";LNUM
5070 IF INT(ABS(LNUM))<>LNUM OR LNUM<2 THEN GOTO 5060
5080 INPUT "INPUT THE SEED";SEED
5090 SEED=INT(ABS(SEED))
5100 GOSUB 5200
5110 CLS:PRINT"SUMMARY STATISTICS FOR THIS SAMPLE:"
5120 PRINT"SAMPLE MEAN(Xbar)=";MEAN
5130 PRINT"SAMPLE STDV.(s)  =";STDV:PRINT
5140 INPUT "INPUT: 1-MENU; 2-MORE UNIFORM RANDOM NUMBERS;
3-EXIT";Q
5150 ON Q GOTO 500,5030,600
5160 END
5180 '****************************************
5185 'RANDOM NUMBER GENERATION AND SUMMARY STATISTICS
5190 '****************************************
5200 FOR I=1 TO SEED
5210  Y=RND(1)
5220 NEXT I
5230 CLS:PRINTTAB(14);"RANDOM NUMBER"
5240 FOR I=1 TO LNUM
5250  MARK=MARK+1
5260  IF MARK>6 THEN CLS:PRINTTAB(14);"RANDOM NUMBER":MARK=1
5270  RAD=INT(RND(1)*(UPLM-LLM+1))+LLM
5280  XSUM=XSUM+RAD
5290  X2SUM=X2SUM+RAD^2
5300  PRINTTAB(13);I;":";RAD
5310  FOR K=1 TO 150:NEXT K
5320 NEXT I
5340 PRINT"RANDOM NUMBER GENERATION COMPLETE!"
5350 MEAN=XSUM/LNUM
5360 STDV=SQR((LNUM*X2SUM-XSUM^2)/(LNUM*(LNUM-1)))
5370 FOR K=1 TO 500:NEXT K
5380 RETURN
21970 '========================================
21975 '========================================
21980 '****************************************
21985 'ERROR TRAP
```

```
21990 '****************************************
22000 PRINT"ERROR";ERR;"IN LINE;"ERL"
22010 PRINT"CONSULT OPERATORS MANUAL"
22020 END
```

```
0 'TITLE: MODULE 3
1 'DATE: 30 APR 85
2 'PURPOSE: THIS PROGRAM IS DESIGNED TO ASSIST THE STUDENT
IN WORKING WITH PROBABILITY APPLICATIONS OF THE; BINOMIAL;
GEOMETRIC; POISSON; AND THE EXPONENTIAL DISTRIBUTIONS.
3 '
4 'A SEPARATE GROUP OF SUBROUTINES IS DEFINED FOR EACH OF
THE ABOVE APPLICATIONS.
5 '
6 'SIGNIFICANT SUBROUTINES WHICH ARE USED JOINTLY BY ALL THE
ABOVE APPLICATIONS INCLUDE:
7 'A FACTORIAL CALCULATION ROUTINE, WHICH CAN WORK
INDEPENDENTLY OR IN CONJUNCTION WITH ANOTHER COMMON ROUTINE
WHICH CALCULATES COMBINATIONS.
8 '
11 'INPUT VARIABLES
12 'BOUNDARY=BOUNDARY VALUE
13 'K1/K2=LOWER/UPPER BOUNDARY (EXPONENTIAL APPLICATION)
14 'LAMBDA=Lambda
15 'MEAN=MEANF OF DATA SET
16 'NTRIALS=NUMBER OF TRIALS
17 'OPER$=OPERATOR (=, <, >, <, =>, <=)
18 'PROB=PROBABILITY
19 'Q=MULTIPURPOSE FLOW CONTROL VARIABLE
20 '
21 'OUTPUT VARIABLES
22 'ANS=DISCRETE ANSWER
23 'FANS=FINAL ANSWER (POISSON APPLICATION)
24 'SANS=SUM OF DISCRETE ANSWERS
25 'S2ANS=ROUNDED VALUE FOR SANS
26 'TNS=ROUNDED VALUE FOR ANS
27 '
28 'INTERNAL VARIABLES
29 'A1NS=DISCRETE ANSWER AT LOWER BOUND
30 'A2NS=DISCRETE ANSWER AT UPPER BOUND
31 'CMB#=SOLUTION TO COMBINATION CALCULATION
32 'K/I = LOOP INDEX
33 'LLIMIT/UPLIMIT = UPPER AND LOWER LIMITS
34 'R/X = SUBSTITUTE VARIABLES FOR LOOP INDICIES
35 'WBOUNDARY=SUBSTITUTE VARIABLE FOR BOUNDARY
36 '
480 '****************************************
485 'TOTAL PROGRAM MENU SELECTION
490 '****************************************
500 ON ERROR GOTO 22000
510 CLS:PRINTTAB(18);"MENU"
520 PRINT"1-BINOMIAL DISTRIBUTION"
530 PRINT"2-GEOMETRIC DISTRIBUTION"
540 PRINT"3-POISSON DISTRIBUTION"
550 PRINT"4-EXPONENTIAL DISTRIBUTION"
```

```
560 PRINT"5-EXIT":PRINT
570 INPUT "SELECTION";Q
580 ON Q GOTO 800,3800,3000,4800,600
590 END
600 MENU
770 '======================================
775 '======================================
780 '**************************************
785 'BINOMIAL DIST. MAIN PROGRAM
790 'INCLUDES DATA INPUT/OUTPUT
795 '**************************************
800 CLS:PRINT"*********BINOMIAL DISTRIBUTION**********"
810 PRINT"METHOD: SINCE THE BINOMIAL DIST. IS     DISCRETE,
THE PROBABILITY BETWEEN TWO   POINTS IS MERELY THE SUM OF
THE         PROBABILITIES OF EACH OF THE INDIVIDUAL INTERIOR
POINTS."
820 INPUT "PUSH ENTER TO CONTINUE!";Q
830 CLS:PRINT"METHOD (continued):THIS PROGRAM
CALCULATES AND DISPLAYS THE PROBABILITY OF EACH OF THE
INDIVIDUAL POINTS AND    THEN SUMS THESE VALUES TO OBTAIN A
FINAL SOLUTION."
840 PRINT"(USE TABLE V TO SPEED YOUR CALCULATIONS)"
850 INPUT"PUSH ENTER TO CONTINUE!";Q
860 CLS:CLEAR:DEFSNG A-Z:ON ERROR GOTO 22000
865 PRINT"STEP 1. DETERMINE FROM THE PROBLEM THE  NUMBER OF
REQUIRED TRIALS (sample size).":PRINT
870 INPUT "INPUT THE NUMBER OF TRIALS (N)";NTRIALS:N=NTRIALS
880 CLS:PRINT"STEP 2. DETERMINE THE PROBABILITY OF A  SINGLE
OCCURRENCE.(almost always clearly stated in the
problem)":PRINT
890 INPUT "INPUT PROBABILITY(in decimal form)";PROB
895 IF PROB >1 OR PROB<0 THEN GOTO 890
900 CLS:PRINT"STEP 3. DETERMINE THE BOUNDARY VALUE.    (EXP:
hoplifters 3 will   get caught; less than 3; 3 or more...
In each case 3 is the BOUNDARY VALUE!.)":PRINT
910 INPUT "INPUT THE BOUNDARY VALUE";BOUNDARY
915 IF ABS(INT(BOUNDARY))<>BOUNDARY THEN PRINT"BOUNDARY MUST
BE A POSITIVE INTEGER!":GOTO 910
916 IF BOUNDARY>NTRIALS THEN ERROR 60
920 CLS:PRINT"STEP 4. DETERMINE THE IF THE PROBLEM    CALLS
FOR A PROBABILITY <, >, =, <=, or >= THE BOUNDARY
VALUE.":PRINT
930 INPUT "INPUT THE PROPER OPERATOR(<,>,=,<=,>=)";OPER$
935 IF OPER$<>"=" AND OPER$<>"<" AND OPER$<>"<=" AND
OPER$<>"=<" AND OPER$<>">" AND OPER$<>">=" AND OPER$<>"=>"
THEN GOTO 930
940 IF OPER$="=" THEN LLIMIT=BOUNDARY:UPLIMIT=BOUNDARY
950 IF OPER$=">" THEN LLIMIT=BOUNDARY+1:UPLIMIT=NTRIALS
960 IF OPER$="<" THEN LLIMIT=0:UPLIMIT=BOUNDARY-1
970 IF OPER$=">=" OR OPER$="=>" THEN
```

```
LLIMIT=BOUNDRAY:UPLIMIT=NTRIALS
980 IF OPER$="<=" OR OPER$="=<" THEN
LLIMIT=0:UPLIMIT=BOUNDARY
990 GOSUB 1100
1000 PRINT"THE ANSWER IS THE SUM OF THE INDIVIDUAL
PROBABILITIES- IN THIS CASE THIS ="
1010 PRINTSANS;"OR ROUNDED OFF TO";S2ANS:PRINT
1020 INPUT "SELECT: 1-MENU; 2-FOR MORE BINOMIAL
PROBLEMS; 3-EXIT";Q
1030 ON Q GOTO 500,860,600
1040 END
1080 '**************************************
1085 'BINOMIAL CALCULATION
1090 '**************************************
1100 FOR K=LLIMIT TO UPLIMIT
1110  R=K
1120  GOSUB 14100
1130  ANS=CMB#*PROB^K*(1-PROB)^(NTRIALS-K)
1140  SANS=SANS+ANS
1145  TNS=INT(1000*ANS+.5)/1000
1150  IF PROB=.05 OR PROB=.1 OR PROB=.2 OR PROB=.3 OR
PROB=.4 OR PROB=.5 OR PROB=.6 OR PROB=.7 OR PROB=.8 OR
PROB=.9 OR PROB=.95 AND NTRIAL<16 THEN GOSUB 1500
1160  PRINT"FOR K=";K;"THE BIN. PROB.="TNS:FOR
Q=1TO600:NEXTQ
1165  S2ANS=S2ANS+TNS
1166  IF S2ANS>0 THEN ZFLAG$="SET"
1167  IF TNS=0 AND ZFLAG$="SET" THEN GOTO 1180
1170 NEXT K
1180 CLS:PRINT"CALCULATION OF THE INDIVIDUAL BINOMIAL
PROBABILITIES IS COMPLETE!!"
1200 RETURN
1480 '**************************************
1485 'STUDENT INPUT VALUE COMPARISON ROUTINE
1490 '**************************************
1500 CLS:PRINT"INPUT THE BINOMIAL PROBABILITY FOR
N=";NTRAIL;";K(or X)=";K;";WITH AN INDIVIDUAL PROBABILITY
OF";PROB:PRINT "(use table V or formula-4)"
1510 INPUT TPROB
1520 CLS:IF TPROB<>TNS THEN PRINT"WRONG" ELSE PRINT
"CORRECT"
1540 RETURN
2970 '======================================
2975 '======================================
2980 '**************************************
2985 'INTRODUCTION TO THE POISSON DIST.
2990 '**************************************
3000 CLS:PRINT"**********POISSON DISTRIBUTION**********"
3010 PRINT"THE POISSON DISTRIBUTION IS OFTEN USED  TO
ESTIMATE BINOMIAL PROBABILITIES WHEN THE NUMBER OF TRIALS
```

```
(sample size)(N) IS LARGE AND THE PROBABILITY OF A SINGLE
OCCURRENCE (P) IS SMALL."
3020 INPUT "PUSH ENTER TO CONTINUE";Q
3030 CLS:PRINT"MORE FREQUENTLY, THE POISSON DIST. IS   USED
TO OBTAIN PROBABILITY VALUES FOR   RARE EVENTS OVER A PERIOD
OF TIME."
3040 PRINT"METHOD: SINCE THE POISSON DIST. IS      DISCRETE,
THE PROBABILITY BETWEEN TWO   POINTS IS MERELY THE SUM OF
THE         PROBABILITIES OF EACH OF THE INTERIOR
POINTS!";
3050 INPUT "PUSH ENTER TO CONTINUE";Q
3060 CLS:PRINT"METHOD (continued): THIS PROGRAM
CALCULATES AND DISPLAYS THE PROB. OF    EACH OF THE INTERIOR
POINTS, THEN SUMS  THESE VALUES TO OBTAIN A SOLUTION."
3070 PRINT"(THE CALCULATIONS MAKE USE OF FORMULA-8)"
3073 '****************************************
3075 DATA INPUT/OUTPUT
3077 '****************************************
3080 CLEAR:DEFSNG A-Z:ON ERROR GOTO 22000
3090 INPUT "INPUT: 1-TO ESTIMATE BINOMIAL PROB.:    2-ALL
OTHER POISSON APPLICATIONS";Q
3100 IF Q<>1 AND Q<>2 THEN GOTO 3090
3110 IF Q=1 THEN GOSUB 3500 ELSE GOSUB 3550
3120 CLS:PRINT"STEP 2. DETERMINE THE BOUNDARY VALUE.
(EXP: ... what is the probability of    receiving more than
6 bad checks in a   day. In this case the BOUNDARY VALUE =
6)":PRINT
3130 INPUT "INPUT THE BOUNDARY VALUE(K)";BOUNDARY
3140 IF BOUNDARY <O THEN PRINT"BOUNDARY MUST BE >0":GOTO
3130
3150 CLS:PRINT"STEP 3. DETERMINE IF THE PROBLEM CALLS  FOR A
PROBABILITY >, <, =, <=,or >= THE BOUNDARY VALUE.":PRINT
3160 INPUT "INPUT THE PROPER OPERATOR (<,>,=,<=,>=)";OPER$
3170 IF OPER$<>"=" AND OPER$<>"<" AND OPER$<>">" AND
OPER$<>"<=" AND OPER$<>"=<" AND OPER$<>">=" AND OPER$<>"=>"
THEN PRINT"REENTER OPERATOR!":GOTO 3160
3180 CLS:GOSUB 3600
3190 IF OPER$=">" OR OPER$=">=" OR OPER$="=>" THEN PRINT "TO
OBTAIN THE ANSWER SUBTRACT P(";K-1;")=";SANS;"FROM
ONE":FANS=1-SANS ELSE FANS=SANS
3200 PRINT"THE PROBABILITY FOR K:";OPER$;BOUNDARY;"IS
=";FANS
3210 FOR Q=1TO1200:NEXTQ
3220 INPUT "SELECT: 1-MENU; 2-MORE POISSON PROBLEMS;
3-EXIT";Q
3230 ON Q GOTO 500,3080,600
3240 END
3480 '****************************************
3485 'ADDITIONAL DATA INPUT FOR BINOMIAL ESTIMATION
3490 '****************************************
```

```
3500 CLS:PRINT"STEP 1. DETERMINE FROM THE PROBLEM THE
NUMBER OF REQUIRED TRIALS (sample size).":PRINT
3510 INPUT "INPUT THE NUMBER OF TRIALS (N)";NTRIALS
3520 IF ABS(INT(NTRIALS))<>NTRIALS THEN PRINT"MUST BE AN
INTEGER VALUE!":GOTO 3510
3530 CLS:PRINT"STEP 1(a). DETERMINE THE PROBABILITY OF A
SINGLE OCCURRENCE.":PRINT
3535 INPUT "INPUT PROBABILITY(in decimal form)";PROB
3540 IF PROB>1 OR PROB<0 THEN GOTO3535
3545 LAMBDA=NTRIALS*PROB:RETURN
3546 '****************************************
3547 'LAMBDA INPUT
3548 '****************************************
3550 CLS:PRINT"STEP 1. DETERMINE THE VALUE OF LAMBDA:  (This
value may be clearly stated in the problem or you may have
to derive it.)(Remember:LAMBDA=RATE*TIME)(Units of measure
must be the same)":PRINT
3560 INPUT "INPUT LAMBDA";LAMBDA
3570 RETURN
3580 '****************************************
3585 'POISSON CALCULATIONS
3590 '****************************************
3600 IF OPER$="="THEN LLIMIT=BOUNDARY:UPLIMIT=BOUNDARY
3610 IF OPER$=">"THENPRINT"FIRST COMPUTE P(K)(<=) AND
SUBTRACT THIS FROM ONE TO OBTAIN
Q(K)(>)!":LLIMIT=0:UPLIMIT=BOUNDARY
3620 IF OPER$=">=" OR OPER$="=>"THEN PRINT"FIRST COMPUTE
P(K-1)(<=) AND SUBTRACT IT FROM ONE TO OBTAIN Q(K-1)(>)!"
LLIMIT=0:UPLIMIT=BOUNDARY-1
3630 IF OPER$="<"THEN LLIMIT=0:UPLIMIT=BOUNDARY-1
3640 IF OPER$="<=" OR OPER$="=<" THEN
LLIMIT=0:UPLIMIT=BOUNDARY
3645 FOR K=1TO 1200:NEXT K
3650 FOR K=LLIMIT TO UPLIMIT
3660  X=K:GOSUB 14000
3670  ANS=(LAMDA^K*EXP(-LAMDA))/Y#
3680  SANS=ANS+SANS
3690  PRINT"K=";K;":f(K)=";ANS:FOR Q=1TO400:NEXT Q
3700 NEXT K
3710 RETURN
3770 '========================================
3775 '========================================
3780 '****************************************
3785 'INTRODUCTION TO THE GEOMETRIC DIST.
3790 '****************************************
3800 CLS:PRINT"*********GEOMETRIC DISTRIBUTION*********";
3810 PRINT"THE GEO. DIST. CAN BE USED TO DETERMINE THE PROB.
OF: OBTAINING A DESIRED RESULT ON THE Kth TRIAL-f(K); A
DESIRED RESULT OCCURRING WITHIN K NUMBER OF TRIALS-P(K);
MORE THAN K NUMBER OF TRIALS ARE      NEEDED TO OBTAIN THE
```

```
RESULT- Q(K)."
3820 INPUT "PUSH ENTER TO CONTINUE";Q
3830 CLS:PRINT"METHOD: WHILE THE GEO. DIST. IS DISCRETE, THE
VARIOUS PROBABILITY VALUES CAN BE CALCULATED DIRECTLY USING
THE FOLLOWING FORMULAS.(Derived from page 218)"
3840 PRINT"f(K)= p(1-p)^(K-1) (Formula-5)"
3850 PRINT"P(K)= 1-(1-p)^K (Formula-6)"
3860 PRINT"Q(K)= (1-p)^K (reliability) (Formula-7)"
3870 INPUT "PUSH ENTER TO CONTINUE";Q
3880 '****************************************
3885 'DATA INPUT/OUTPUT
3890 '****************************************
3900 CLS:CLEAR:DEFSNG A-Z:ON ERROR GOTO 22000
3905 PRINT"STEP 1. DETERMINE THE PROBABILITY OF    SUCCESS.
(Remember, 'SUCCESS' is    defined in the context of a
problem and may be the occurrence of a 'bad' event!)":PRINT
3910 INPUT "INPUT PROBABILITY (in decimal form)";PROB
3920 IF PROB>1 OR PROB<0 THEN PRINT"PROB. MUST BE A POSITIVE
DECIMAL VALUE  BETWEEN 0 AND 1!":GOTO 3910
3930 CLS:PRINT"STEP 2. DETERMINE THE BOUNDARY VALUE.    (EXP
efore the 5th trial?... In each case the boundary value is
5!)":PRINT
3940 INPUT "INPUT THE BOUNDARY
VALUE(K)";BOUNDARY:WBOUNDARY=BOUNDARY
3950 IF BOUNDARY <1 THEN PRINT"BOUNDARY VALUE MUST BE
>=1":GOTO 3940
3960 CLS:PRINT"STEP 3. DETERMINE IF THE PROBLEM CALLS  FOR A
PROBABILITY =, <, >, <=,  OR >=   THE BOUNDARY
VALUE(K).":PRINT
3970 INPUT "INPUT THE PROPER OPERATOR(=,<,>,<=,>=)";OPER$
3980 IF OPER$<>"="AND OPER$<>"<"AND OPER$<>"<="AND
OPER$<>"=<"AND OPER$<>">"AND OPER$<>">="AND OPER$<>"=>" THEN
PRINT"TRY AGAIN":GOTO 3970
3990 CLS:PRINT"STEP 4. ";
4000 IF OPER$=">=" OR OPER$="=>" THEN PRINT"SINCE THE
OPERATOR IS ";OPER$;" YOU NEED TO CALCULATE
Q(";BOUNDARY-1;")(or Q(K-1))":WBOUNDARY=BOUNDARY-1:GOSUB
4450
4010 IF OPER$=">" THEN PRINT"SINCE THE OPERATOR IS ";OPER$;"
YOU NEED TO CALCULATE Q(";BOUNDARY;")":GOSUB 4450
4020 IF OPER$="<=" OR OPER$="=<" THEN PRINT"SINCE THE
OPERATOR IS ";OPER$;" YOU NEED TO CALCULATE
P(";BOUNDARY;")":GOSUB 4500
4030 IF OPER$="<" THEN PRINT"SINCE THE OPERATOR IS ";OPER$;"
YOU NEED TO CALCULATE P(";BOUNDARY-1;")(or
P(K-1))":WBOUNDARY=BOUNDARY-1:GOSUB 4500
4040 IF OPER$="=" THEN PRINT"SINCE THE OPERATOR IS ";OPER$;"
YOU NEED TO CALCULATE f(";BOUNDARY;")(or f(K))":GOSUB 4550
4050 PRINT:INPUT "PUSH ENTER FOR COMPUTER SOLUTION!";Q
4060 CLS:PRINT"THE PROBABILITY FOR K:";OPER$;WBOUNDARY;"IS
```

```
=";ANS:PRINT
4070 INPUT "SELECT: 1-MENU; 2-MORE GEOMETRIC
PROBLEMS; 3-EXIT";Q
4080 ON Q GOTO 500,3900,600
4090 END
4430 '****************************************
4435 'GEOMETRIC CALCULATAIONS
4440 '****************************************
4450 ANS=(1-PROB)^WBOUNDARY:RETURN
4500 ANS=1-(1-PROB)^WBOUNDARY:RETURN
4550 ANS=PROB*(1-PROB)^(WBOUNDARY-1):RETURN
4770 '========================================
4775 '========================================
4780 '****************************************
4785 'EXPONENTIAL DIST. DRIVER
4790 '****************************************
4800 CLEAR:DEFSNG A-Z:ON ERROR GOTO 22000:GOSUB 5000
4810 IF OPER$="B" OR OPER$="b" THEN GOSUB 5500 ELSE GOSUB
5100
4820 INPUT "SELECT: 1-MENU; 2-MORE EXPONENTIAL
PROBLEMS; 3-EXIT";Q
4830 ON Q GOTO 500,4800,600
4840 END
4980 '****************************************
4985 'PROBLEM SET-UP AND DATA INPUT
4990 '****************************************
5000 CLS:PRINT"********EXPONENTIAL DISTRIBUTION********"
5010 PRINT"STEP 1. DETERMINE IF THE PROBLEM ASKS  FOR A
PROBABILITY < OR > A GIVEN VALUE; OR FOR A PROBABILITY
BETWEEN TWO GIVEN  VALUES!
5020 INPUT "INPUT <, >, OR B-(for between two given
values)";OPER$
5030 IF OPER$<>">" AND OPER$<>"<" AND OPER$<>"B" AND
OPER$<>"b" THEN CLS:PRINT"INPUT MUST BE A <,>, OR B:TRY
AGAIN":GOTO 5020
5040 CLS:PRINT"STEP 2. INPUT MU (mean)."
5050 PRINT"REMEMBER: LAMDA=FAILURE RATE"
5060 PRINTTAB(10);"MU=1/LAMDA (or mean time to
failure)":PRINT
5070 INPUT MEAN
5080 CLS:PRINT"STEP 3. DETERMINE THE BOUNDARY VALUE (or
values for the 'between' case).":PRINT
5090 RETURN
5093 '****************************************
5095 'EXPONENTIAL CALCULATIONS AND OUTPUT (for < or > case)
5097 '****************************************
5100 INPUT "INPUT THE BOUNDARY VALUE(K)!";BOUNDARY
5110 CLS:PRINT"STEP 4. COMPUTE THE PROBABILITY. (using
formula-9)":PRINT
5120 PRINT"REMEMBER: SINCE THE OPERATOR IS ";OPER$;" WE
```

```
NEED TO COMPUTE:"
5130 IF OPER$="<" THEN PRINT"P(K) OR
P(";BOUNDARY;")":ANS=1-EXP(-BOUNDARY/MEAN) ELSE PRINT"Q(K)
OR Q(";BOUNDARY;")":ANS=EXP(-BOUNDARY/MEAN)
5140 INPUT "PUSH ENTER TO CONTINUE!";Q
5150 CLS:IF OPER$="<" THEN PRINT"P(";BOUNDARY;") =";ANS ELSE
PRINT"Q(";BOUNDARY;") =";ANS:PRINT
5160 IF OPER$=">" THEN PRINT"NOTE: Given a failure rate,
Q(K) is      often described and the reliability      (usually
expressed as a %)":PRINT
5170 RETURN
5480 '****************************************
5485 'EXPONENTIAL CALCULATIONS AND OUTPUT (for the between
case)
5490 '****************************************
5500 INPUT "INPUT THE LOWER BOUNDARY(K1)";K1
5510 INPUT "INPUT THE UPPER BOUNDARY(K2)";K2
5530 CLS:PRINT"STEP 4. COMPUTE P(K1).                    (use
P(x)=1-exp(-X/MU)":PRINT"(remember K1=";K1;")"
5540 A1NS=1-EXP(-K1/MEAN)
5550 PRINT"COMPUTER SOLUTION FOR P(";K1;") =";A1NS:PRINT
5555 INPUT "PUSH ENTER TO CONTINUE!";Q
5560 CLS:PRINT"STEP 5. COMPUTE P(K2).":PRINT
5570 A2NS=1-EXP(-K2/MEAN)
5580 PRINT"COMPUTER SOLUTION FOR P(";K2;") =";A2NS:PRINT
5585 INPUT "PUSH ENTER TO CONTINUE!";Q
5590 CLS:PRINT"STEP 6. SUBTRACT P(K1) FROM P(K2) TO
OBTAIN THE PROBABILITY BETWEEN K1 AND K2.":PRINT
5600 PRINT"COMPUTER SOLUTION:"
5610 PRINT A2NS;"-";A1NS;"=";A2NS-A1NS
5620 RETURN
13970 '========================================
13975 '========================================
13980 '****************************************
13985 'FACTORIAL CALCULATIONS
13990 '****************************************
14000 Y#=X
14005 IF INT(X)<>X THEN ERROR 61
14010 IF X>48 OR X<0 THEN ERROR 62
14015 IF X=1 OR X=0 THEN Y#=1:RETURN
14020 FOR I=ABS(X)-1 TO 2 STEP-1
14030  Y#=I*Y#
14040 NEXT I
14050 RETURN
14080 '****************************************
14085 'COMBINATIONS CALCULATIONS
14090 '****************************************
14100 N=NTRIALS:IF N<R THEN ERROR 60
14110 IF N-R<R THEN R=N-R
14120 J=N
```

```
14130 IF R=0 THEN J=1:GOTO 14200
14140 IF R=1 THEN GOTO 14200
14150 FOR I=1 TO R-1
14160  N=N-1
14170  J=J*N
14180 NEXT I
14200 X=R
14210 GOSUB 14000
14220 CMB#=J/Y#
14230 RETURN
21980 '****************************************
21985 'ERROR TRAP
21990 '****************************************
22000 IF ERR=60 THEN PRINT"N MUST BE >=R"
22010 IF ERR=61 THEN PRINT"FACTORIALS CAN ONLY BE COMPUTED
FOR      INTEGER VALUES."
22020 IF ERR=62 OR ERR=6 THEN PRINT"THE COMPUTED NUMBER IS
TOO LARGE, AND    OUT OF RANGE OF THIS SYSTEM, SORRY!"
22030 IF ERR=11 THEN PRINT"DIVISION BY ZERO ERROR. CHECK
YOUR INPUTS!!"
22040 PRINT"ERROR";ERR;"IN LINE";ERL
22050 PRINT"CONSULT OPERATORS MANUAL!!!"
22060 END
```

```
0 'TITLE: MODULE 4
1 'DATE: 30 APR 85
2 'PURPOSE: THIS PROGRAM IS DESIGNED TO ASSIST THE STUDENT
IN LEARNING STANDARD PROBLEM SOLVING TECHNIQUES WHICH CAN BE
APPLIED TO THE FOLLOWING TYPES OF STATISTICAL PROBLEMS:
3 'APPLICATIONS OF THE CENTRAL LIMIT THEOREM.
4 'APPLICATIONS OF THE NORMAL DISTRIBUTION.
5 '
6 'A SEPARATE GROUP OF SUBROUTINES IS DEFINED FOR EACH
APPLICATION.
7 '
8 'SIGNIFICANT SUBROUTINES WHICH ARE JOINTLY USED BY BOTH
APPLICATIONS INCLUDE:
9 'A DATA TABLE ROUTINE WHICH PROVIDES  STANDARD Z VALUES;
AND A ROUTINE WHICH COMPARES A STUDENT'S INPUT FOR Z-VALUES
TO THE DATA TABLE.
10 '
11 'INPUT VARIABLES:
12 'OPER$=OPERATOR (used in App. of Normal)
13 'PMEAN=POPULATION MEAN
14 'PROBVLUE=Z-ALPHA VALUE
15 'Q=USED TO SELECT PROGRAM PATH
16 'SANS=STUDENT ANSWER
17 'SNUM=SAMPLE SIZE
18 'STDV=STANDARD DEVIATION
19 'X=DIFFERENCE BETWEEN SAMPLE MEAN AND POP. MEAN / LIMIT
VALUE
20 'ZSCRE=STUDENT GUESS FOR Z-SCORE
21 '
22 'OUTPUT VARIABLES
23 'ANS=ANSWER
24 'TL=TABLE VALUE
25 '
26 'INTERNAL VARIABLES
27 'FLAG$=FLOW CONTROL
28 'PTR%=POINTER (in table)
29 'TABLE$=DATA TABLE VALUES
30 'TSVLUE=TABLE VALUE STRING
31 'TY:W = INTERMEDIATE VALUES
32 'XL:XU = SUBSTITUTE VARIABLES FOR ZVLUE
33 'XP:YP = SUBSTITUTE VARIABLES FOR PROBVLUE
34 'ZVLUE=Z-SCORE(computer generated)
35 '
85 '****************************************
90 'TOTAL PROGRAM MENU SELECTION
95 '****************************************
100 CLS:PRINTTAB(18);"MENU"
110 PRINT"1-CENTRAL LIMIT THEOREM"
120 PRINT"2-APPLICATIONS OF THE NORMAL DISTO."
130 PRINT"3-EXIT"
```

```
140 PRINT:PRINT
150 INPUT "SELECTION";Q
160 ON Q GOTO 500,2500,180
170 END
180 MENU
470 '========================================
475 '========================================
480 '****************************************
485 'C.L.THEOREM INITIALIZATION ROUTINE
490 '****************************************
500 CLEAR:GOSUB 20000
510 GOSUB 1000
520 INPUT "SELECT: 1-MENU; 2-MORE C.L.T. PROBLEMS; 3-EXIT";Q
530 ON Q GOTO 100,500,180
540 END
980 '****************************************
985 'C.L.THEOREM MAIN PROGRAM
990 '****************************************
1000 CLS:PRINT"***********CENTRAL LIMIT THEOREM********"
1010 PRINT"THIS USE OF THE C.L.T. COMPUTES THE
PROBABILITY OF THE DIFFERENCE BETWEEN   THE SAMPLE MEAN AND
THE POPULATION MEAN BEING LESS THAN A GIVEN VALUE 'X'."
1020 PRINT:INPUT "INPUT 1-TO CONTINUE;2-TO QUIT";Q
1025 IF Q<>1 THEN GOTO 100
1030 CLS:PRINT"STEP 1."
1040 PRINT"DETERMINE THE DIFFERENCE BETWEEN THE
POPULATION MEAN AND THE SAMPLE MEAN TO  BE TESTED.":PRINT
1050 INPUT"INPUT THIS DIFFERENCE (X)";X
1060 CLS:PRINT"STEP 2."
1070 PRINT"DETERMINE FROM THE PROBLEM THE SAMPLE   SIZE AND
THE POPULATION STANDARD DEV.":PRINT
1080 INPUT "INPUT SAMPLE SIZE";SNUM
1083 IF ABS(INT(SNUM))<>SNUM THEN PRINT"MUST BE A POSITIVE
INTEGER!":GOTO1080
1085 INPUT "INPUT POPULATION STD. DEV.";STDV
1087 IF ABS(STDV)<>STDV OR STDV=0 THEN PRINT"MUST BE
POSITIVE!":GOTO 1085
1090 CLS:PRINT"STEP 3."
1100 PRINT"USING THE VALUES OF 'X', STD. DEV. AND  THE
SAMPLE SIZE, COMPUTE THE Z-SCORE    (STANDARD UNITS).
(Z-SCORE = X/(POP.    STD. DEV./SQR(SAM. SIZE))":PRINT
1110 ZVLUE=X/(STDV/SQR(SNUM))
1120 INPUT "INPUT Z-SCORE";ZSCRE
1130 CLS:IF ABS(ZSCRE-ZVLUE)>.01 THEN PRINT"WRONG" ELSE
PRINT"CORRECT"
1140 PRINT"USING"
1150 PRINT"X=";X
1160 PRINT"SAMPLE SIZE=";SNUM
1170 PRINT"POP. STD. DEV.=";STDV
1180 PRINT"THE Z-SCORE COMPUTES TO BE";ZVLUE"
```

```
1190 FOR I=1 TO 1600:NEXT I
1200 CLS:PRINT"STEP 4."
1210 PRINT"USE TABLE I TO FIND THE PROPER VALUE
ASSOCIATED WITH A Z-SCORE=";ZVLUE:PRINT
1220 INPUT "INPUT THIS VALUE";PROBVLUE
1230 GOSUB 15000:REM ***RETRIEVE Z-ALPHA VALUE***
1240 FOR I=1 TO 800:NEXT I
1250 CLS:PRINT"STEP 5."
1260 PRINT"REMEMBER, THE OBTAINED TABLE VALUE
REPRESENTS THE (PROBABILITY) AREA       BETWEEN 0
AND";ZVLUE;"."
1270 PRINT"IN ORDER TO ACCOUNT FOR THE AREA BETWEEN
-";ZVLUE;"AND 0 WE MUST MULTIPLY THE   TABLE VALUE BY TWO."
1280 INPUT "PUSH ENTER TO CONTINUE";Q
1290 CLS:PRINT PROBVLUE;"TIMES TWO IS =";PROBVLUE*2
1300 PRINT"THEREFORE, THIS IS THE PROBABILITY THAT THE
SAMPLE MEAN WILL DIFFER FROM THE    POP.MEAN BY LESS
THAN";X:PRINT
1310 RETURN
2470 '=======================================
2475 '=======================================
2480 '***************************************
2485 APP. OF THE NORMAL DRIVER
2490 '***************************************
2500 CLEAR:GOSUB 20000
2510 GOSUB 3000
2520 IF OPER$="B" OR OPER$="b" THEN GOSUB 4000 ELSE GOSUB
5000
2530 INPUT "SELECT; 1-MENU;2-MORE APP. OF THE NORMAL;
3-EXIT";Q
2535 ON Q GOTO 100,2500,180
2540 END
2980 '***************************************
2985 'BASIC DATA INPUT (for problem definition)
2990 '***************************************
3000 CLS:PRINT"APPLICATIONS OF THE NORMAL DISTRIBUTION"
3005 PRINT"***************************************"
3010 PRINT"THIS PROGRAM IS DESIGNED TO AID IN      COMPUTING
THE AREA (AND THUS, THE        PROBABILITY VALUE) UNDER THE
NORMAL     CURVE BETWEEN VARIOUS POINTS.":PRINT
3020 INPUT "ENTER 1-TO CONTINUE;2-TO EXIT";Q
3030 IF Q<>1 THEN GOTO 100
3040 CLS:PRINT"STEP 1."
3050 PRINT"DETERMINE IF THE PROBLEM CALLS FOR      FINDING A
PROBABILITY > OR < A GIVEN     VALUE(X); OR BETWEEN TWO GIVEN
VALUES    (X and Y)":PRINT
3060 INPUT "INPUT <;>; OR B (for between two values)";OPER$
3065 IF OPER$<>"<" AND OPER$<>">" AND OPER$<>"B" AND
OPER$<>"b" THEN GOTO 3060
3070 CLS:PRINT"STEP 2."
```

```
3080 PRINT"DETERMINE THE POPULATION MEAN AND STD. DEV.
(usually given).":PRINT
3090 INPUT "ENTER THIS MEAN";PMEAM
3095 INPUT "ENTER THE STD.DEV.";STDV
3097 IF STDV<=0 THEN PRINT"MUST BE POSITIVE!":GOTO 3095
3100 RETURN
3980 '****************************************
3985 'DATA INPUT FOR COMPUTATION OF PROB BETWEEN TWO POINTS
3990 '****************************************
4000 CLS:PRINT"TO COMPUTE THE PROBABILITY BETWEEN TWO
VALUES (X and Y) YOU NEED TO CONVERT X  and Y TO STANDARD
UNITS (Z-SCORES) AND  THEN FIND THE CORRESPONDING TABLE I
ENTRIES"
4010 PRINT"STEP 3."
4020 PRINT"LETS START WITH THE LOWER LIMIT X."
4025 FLAG$="LOWER"
4030 FOR J=1 TO 2
4040     INPUT "ENTER THE LIMIT VALUE";X
4050     CLS:PRINT"STEP 4"
4060     PRINT"CONVERT THIS VALUE TO A Z-SCORE.
(Z-SCORE = (X-POP.MEAN)/STD.DEV.)":PRINT
4070     GOSUB 6070:REM ***CALCULATE Z-SCORE
4210     IF FLAG$<>"UPPER" THEN XP=PROBVLUE:XL=ZVLUE ELSE
YP=PROBVLUE:XU=ZVLUE
4230     IF FLAG$<>"UPPER" THEN CLS:PRINT"NOW LETS DO THE
SAME THING FOR THE UPPER VALUE Y":PRINT"STEP 3."
4245     FLAG$="UPPER"
4255 NEXT J
4260 CLS:PRINT"STEP 6."
4270 PRINT"USING THE VALUES OBTAINED FROM TABLE I, COMPUTE
THE AREA BETWEEN";XL;"AND";XU;"."
4280 PRINT"DRAWING A PICTURE WILL HELP!(REMEMBER   THE TABLE
VALUES GIVE THE AREA BETWEEN  0 AND THE GIVEN LIMIT.)
4290 INPUT "PUSH ENTER TO CONTINUE";Q
4300 CLS:PRINT"TABLE VALUE CORRESPONDING TO";XL;"=";XP
4310 PRINT"TABLE VALUE CORRESPONDING TO";XU;"=";YP:PRINT
4320 INPUT "COMPUTE THE PROBABILITY AND  INPUT THE
ANSWER!";SANS
4330 IF SGN(XL) XOR SGN(XU) THEN ANS=XP+YP ELSE
ANS=ABS(XP-YP)
4340 CLS:IF ABS(SANS-ANS)>.0001 THEN PRINT"WRONG" ELSE
PRINT"CORRECT"
4350 PRINT"THE AREA BETWEEN";XL;"AND";XU;"=";ANS;", (which
corresponds to the probability).":PRINT
4360 RETURN
4980 '****************************************
4985 'DATA INPUT FOR COMPUTATION OF PROB. > OR < A GIVEN
VALUE.
4990 '****************************************
5000 CLS:PRINT"STEP 3."
```

```
5010 PRINT"EXTRACT THE VALUE OF THE LIMIT 'X' FROM THE
PROBLEM":PRINT
5020 INPUT "ENTER LIMIT VALUE 'X'";X
5030 CLS:PRINT"STEP 4."
5040 PRINT"CONVERT THE GIVEN LIMIT VALUE 'X' TO A  Z-SCORE."
5045 PRINT"(Z-SCORE = (X-POP.MEAN)/STD.DEV.)":PRINT
5050 GOSUB 6070:REM ***CALCULATE Z-SCORE
5190 CLS:PRINT"STEP 6."
5200 PRINT"USING THE VALUE OBTAINED FROM THE TABLE, COMPUTE
THE NEEDED PROBABILITY."
5210 PRINT"REMEMBER THAT THE TABLE VALUE";PROBVLUE;"IS  THE
AREA BETWEEN 0 AND";ZVLUE;".";
5220 IF OPER$=">" THEN GOSUB 5500 ELSE GOSUB 5750:REM
***CALCULATE PROB. VALUE***
5230 PRINTCHR$(27)+"pINPUT YOUR ANSWER!"+CHR$(27)+"q";
5233 INPUT SANS
5235 CLS
5240 IF ABS(ANS-SANS)>.0001 THEN PRINT"WRONG" ELSE
PRINT"CORRECT!!"
5250 PRINT"THE PROBABILITY OF OBTAINING A VALUE ";OPER$;"
";X;"IS =";ANS
5260 RETURN
5480 '****************************************
5485 'COMPUTATION OF PROB. > A GIVEN VALUE.
5490 '****************************************
5500 PRINT"SINCE THE OPERATOR IS >, YOU NEED TO DETERMINE
THE AREA UNDER THE CURVE TO THE RIGHT OF";ZVLUE;"."
5510 IF ZVLUE>0 THEN ANS=.5-PROBVLUE ELSE ANS =.5+PROBVLUE
5520 RETURN
5730 '****************************************
5735 'COMPUTATION OF PROB. < A GIVEN VALUE.
5740 '****************************************
5750 PRINT"SINCE THE OPERATOR IS <, YOU NEED TO DETERMINE
THE AREA UNDER THE CURVE TO THE LEFT OF";ZVLUE;"."
5760 IF ZVLUE>0 THEN ANS=.5+PROBVLU ELSE ANS=.5-PROBVLUE
5780 RETURN
5980 '****************************************
5985 'CALCULATION OF Z-SCORE
5990 '****************************************
6070 INPUT "INPUT THE Z-SCORE";ZSCRE
6080 ZVLUE=(X-PMEAN)/STDV
6090 CLS:IF ABS(ZVLUE-ZSCRE)>.01 THEN PRINT"WRONG!!" ELSE
PRINT"CORRECT!!"
6100 PRINT"USING THESE VALUES:"
6110 PRINT"POPULATION MEAN=";PMEAN
6120 PRINT"STD. DEV.=";STDV
6130 PRINT"X=";X
6140 PRINT"THE PROPER Z-SCORE=";ZVLUE
6150 FOR I=1 TO 1600:NEXT I
6160 CLS:PRINT"STEP 5."
```

```
6170 PRINT"FIND THE CORRESPONDING VALUE FROM TABLE I FOR A
Z-SCORE OF";ZVLUE;".":PRINT
6180 INPUT "ENTER THIS VALUE";PROBVLUE
6190 GOSUB 15000:REM ***RETRIEVE Z-ALPHA VALUE***
6200 FOR I= 1 TO 800:NEXT I
6210 RETURN
14980 '****************************************
14985 'CALCULATION OF Z-ALPHA VALUE
14990 '****************************************
15000 IF ABS(ZVLUE)+.005>3.099 THEN TL=.4999:GOTO 15080
15010 TY=INT((ABS(ZVLUE)+.005)*10)
15020 W=(ABS(ZVLUE)+.005)*10-TY
15030 PTR%=((INT(W*10))*5)+1
15060 TSVLUE$=MID$(TABLE$(TY),PTR%,5)
15070 TL=VAL(TSVLUE$)
15080 IF ABS(TL-PROBVLUE)>.00009 THEN PRINT "WRONG TABLE
VALUE!!TRY AGAIN" ELSE PRINT"CORRECT":PROBVLUE=TL:RETURN
15090 INPUT "INPUT YOUR TABLE VALUE AGAIN";PROBVLUE
15100 IF ABS(TL-PROBVLUE)>.00009 THEN PRINT "WRONG
AGAIN!!THE CORRECT VALUE IS";TL ELSE PRINT"RIGHT ON"
15120 PROBVLUE=TL
15130 RETURN
19980 '****************************************
19985 'Z-ALPHA DATA TABLE INPUT
19990 '****************************************
20000 DEFSNG A-Z
20020 ON ERROR GOTO 22000
20030 DIM TABLE$(30)
20040 FOR I=0 TO 30
20050   READ TABLE$(I)
20060 NEXT I
20070
DATA".0000.0040.0080.0120.0160.0199.0239.0279.0319.0359"
20071
DATA".0398.0438.0478.0517.0557.0596.0636.0675.0714.0753"
20072
DATA".0793.0832.0871.0910.0948.0987.1026.1064.1103.1141"
20073
DATA".1179.1217.1255.1293.1331.1368.1406.1443.1480.1517"
20074
DATA".1554.1591.1628.1664.1700.1736.1772.1808.1844.1879"
20075
DATA".1915.1950.1985.2019.2054.2088.2123.2157.2190.2224"
20076
DATA".2257.2291.2324.2357.2389.2422.2454.2486.2517.2549"
20077
DATA".2580.2611.2642.2673.2704.2734.2764.2794.2823.2852"
20078
DATA".2881.2910.2939.2967.2995.3023.3051.3078.3106.3133"
20079
```

```
DATA".3159.3186.3212.3238.3264.3289.3315.3340.3365.3389"
20080
DATA".3413.3438.3461.3485.3508.3531.3554.3577.3599.3621"
20081
DATA".3643.3665.3686.3708.3729.3749.3770.3790.3810.3830"
20082
DATA".3849.3869.3888.3907.3925.3944.3962.3980.3997.4015"
20083
DATA".4032.4049.4066.4082.4099.4115.4131.4147.4162.4177"
20084
DATA".4192.4207.4222.4236.4251.4265.4279.4292.4306.4319"
20085
DATA".4332.4345.4357.4370.4382.4394.4406.4418.4429.4441"
20086
DATA".4452.4463.4474.4484.4495.4505.4515.4525.4535.4545"
20087
DATA".4554.4564.4573.4582.4591.4599.4608.4616.4625.4633"
20088
DATA".4641.4649.4656.4664.4671.4678.4686.4693.4699.4706"
20089
DATA".4713.4719.4726.4732.4738.4744.4750.4756.4761.4767"
20090
DATA".4772.4778.4783.4788.4793.4798.4803.4808.4812.4817"
20091
DATA".4821.4826.4830.4834.4838.4842.4846.4850.4854.4857"
20092
DATA".4861.4864.4868.4871.4875.4878.4881.4884.4887.4890"
20093
DATA".4893.4896.4898.4901.4904.4906.4909.4911.4913.4916"
20094
DATA".4918.4920.4922.4925.4927.4929.4931.4932.4934.4936"
20095
DATA".4938.4940.4941.4943.4945.4946.4948.4949.4951.4952"
20096
DATA".4953.4955.4956.4957.4959.4960.4961.4962.4963.4964"
20097
DATA".4965.4966.4967.4968.4969.4970.4971.4972.4973.4974"
20098
DATA".4974.4975.4976.4977.4977.4978.4979.4979.4980.4981"
20099
DATA".4981.4982.4982.4983.4984.4984.4985.4985.4986.4986"
20100
DATA".4987.4987.4987.4988.4988.4989.4989.4989.4990.4990"
20200 RETURN
21980 '****************************************
21985 'ERROR TRAP
21990 ''****************************************
22000 PRINT "ERROR";ERR;"IN LINE";ERL
22010 PRINT"CONSULT OPERATORS MANUAL!!!"
22020 END
```

```
0 'TITLE: MODULE 5
1 'DATE:  11 APRIL 85
2 'PURPOSE: THIS PROGRAM IS DESIGNED TO ASSIST THE STUDENT
IN LEARNING STANDARD PROBLEM SOLVING TECHNIQUES WHICH CAN BE
APPLIED TO FOLLOWING TYPES OF STATISTICAL PROBLEMS:
3 'HYPOTHESIS TESTING CONCERNING THE MEAN.
4 'DERIVING A CONFIDENCE INTERVAL ABOUT THE MEAN.
5 'DETERMINATION OF THE MAXIMUM ERROR INVOLVED IN AN
ESTIMATE.
6 'DETERMINATION OF THE MINIMUM SAMPLE SIZE NECESSARY TO
ACHIEVE A FIXED LEVEL OF ERROR AT A KNOWN LEVEL OF
SIGNIFICANCE.
7 '
8 'A SEPARATE GROUP OF BASIC ROUTINES IS DEFINED FOR EACH OF
THE ABOVE APPLICATIONS.
9 'SIGNIFICANT SUBROUTINES WHICH ARE JOINTLY USED BY ALL
APPLICATIONS INCLUDE; A DATA TABLE ROUTINE WHICH PROVIDES
T-ALPHA VALUES; AND A ROUTINE WHICH CHECKS THE STUDENT'S
INPUT VALUE OF T-ALPHA AGAINST THE TABLE VALUE.
10 '
11 'INPUT VARIABLES:
12 'NHYP=VALUE OF THE NULL HYPOTHESIS
13 'OPER$=OPERATOR OF THE ALTERNATE HYPOTHESIS
14 'LSIG=LEVEL OF SIGNIFICANCE
15 'SNUM=SAMPLE SIZE
16 'TALPHA=T-ALPHA
17 'STDV=STANDARD DEVIATION
18 'SMEAN=SAMPLE MEAN
19 'DFRD=DEGREES OF FREEDOM
20 'DCNF=DEGREES OF CONFIDENCE
21 '
22 'OUTPUT VARIABLES:
23 'TSTAT=TEST STATISTIC
24 'FLAG$=ACCEPT/REJECT CRITERIA
25 'TL=TABLE VALUE FOR T-ALPHA
26 'MER=MAX. ERROR OF ESTIMATE
27 'LFT=LOWER CI LIMIT
28 'RGTH=UPPER CI LIMIT
29 '
30 'INTERNAL VARIABLES
31 'PTR%=TABLE POINTER
32 'TNUM=SNUM (sample size)
33 'R%:S2:A2 = INTERMEDIATE VALUES
85 '****************************************
90 'TOTAL PROGRAM MENU SELECTION
95 '****************************************
100 CLS:PRINTTAB(18);"MENU"
105 PRINT"ITEM #"
110 PRINTTAB(2);"1      MAX ERROR OF ESTIMATE"
115 PRINTTAB(2);"2      DETERMINE SAMPLE SIZE"
```

```
120 PRINTTAB(2);"3     CONFIDENCE INTERVAL ABOUT MEAN"
125 PRINTTAB(2);"4     HYPOTHESIS TESTING OF MEAN"
130  PRINTTAB(2)"5     EXIT"
135 INPUT "SELECTION #";Q
140 ON Q GOTO 2050,3100,1100,150,145
145 MENU
146 '*****************************************
147 'HYPOTHESIS TESTING DRIVER
148 '*****************************************
150 CLS:PRINT "***HYPOTHESIS TESTING CONCERNING
MEANS**":PRINT
155 PRINT "VALUES NEEDED: NULL HYP. VALUE; SAMPLE  MEAN;
STD. DEV.; SIZE OF SAMPLE."
160 PRINT:PRINT:INPUT "PUSH ENTER TO CONTINUE";Q
165 CLEAR:CLS
170 GOSUB 20000
175 GOSUB 300
180 GOSUB 440
185 GOSUB 500
190 GOSUB 600
195 PRINT:INPUT "SELECT: 1-MORE HYP. TESTING; 2-MENU;
3-EXIT";Q
200 ON Q GOTO 165,100,145
205 END
285 '*****************************************
290 'DATA INPUT ROUTINE
295 '*****************************************
300 PRINT"************SOLUTION STEPS*************"
310 PRINT"STEP 1. FORMULATE THE NULL HYPOTHESIS."
320 PRINT:INPUT "INPUT VALUE OF THE NULL HYP.";NHYP
325 CLS:PRINT "STEP 2. FORMULATE THE ALTERNATE HYPOTHESIS."
330 PRINT:INPUT "IS THE OPERATOR OF THE ALT. HYP. < (less
than), > (greater than), OR <> (not equal to)";OPER$
331 IF OPER$<>">" AND OPER$<>"<" AND OPER$<>"<>" AND
OPER$<>"><"  THEN GOTO 330
335 IF OPER$="<>" OR OPER$="><" THEN R%=2 ELSE R%=1
340 CLS:PRINT "BASED ON THE OPERATOR OF THE ALT. HYP. IS
THIS A 1 OR 2 TAILED TEST?"
345 INPUT Q%:CLS
346 IF Q%<>1 AND Q%<>2 THEN GOTO 340
350 IF Q%=R% THEN PRINT "CORRECT" ELSE PRINT
"SORRY-INCORRECT"
355 IF OPER$="<>" OR OPER$="><" THEN PRINT "<> (not equal
to) INDICATES A TWO TAILED TEST" ELSE PRINT "< OR > INDICATE
A ONE TAILED TEST"
360 FOR I= 1 TO 1200:NEXT I:CLS
365 PRINT "3. DETERMINE THE LEVEL OF SIGNIFICANCE (ALPHA) AS
STATED IN THE PROBLEM"
370 PRINT:INPUT "INPUT LEVEL OF SIG. (in decimal
form)";LSIG:S2=LSIG/R%
```

```
371 IF LSIG>1 OR LSIG<0  THEN PRINT"TRY AGAIN":GOTO 370
375 INPUT "HOW LARGE IS THE SAMPLE";SNUM
380 IF ABS(INT(SNUM))<>SNUM THEN GOTO 375
390 CLS:PRINT"4. DETERMINE FROM THE TABLE THE PROPER
T-ALPHA. REMEMBER TO USE THE RIGHT DEGREES OF FREEDOM
(N-1)":PRINT
395 PRINT "FIND THE T-ALPHA CORRESPONDING
TO";LSIG/R%;"WITH";SNUM-1;"DEGREES OF FREEDOM"
400 INPUT "INPUT T-ALPHA VALUE";TALPHA
405 GOSUB 15000:REM COMPARE INPUT VALUE TO TABLE VALUE
410 INPUT "STANDARD DEV.";STDV
412 IF STDV<0 THEN PRINT"STDV MUST BE POSITIVE":GOTO 410
415 INPUT "SAMPLE MEAN";SMEAN:CLS
420 PRINT"STEP 5. COMPUTE THE TEST STATISTIC. PUSH ENTER AND
THE COMPUTER WILL DO IT FOR YOU."
425 INPUT Q
430 CLS:RETURN
435 '****************************************
436 'COMPUTATION OF THE TEST STATISTIC
437 '****************************************
440 TSTAT=(SMEAN-NHYP)/(STDV/SQR(SNUM))
445 CLS:PRINT"GIVEN THE FOLLOWING INPUTS:"
450 PRINT"NULL HYP =";NHYP;"   SAM SIZE =";SNUM
455 PRINT"STD.DEV. =";STDV;"   SAM MEAN =";SMEAN
460 PRINT "THE TEST STATISTIC IS =";TSAT
465 FOR I=1 TO 1500:NEXT I
470 RETURN
475 '****************************************
480 'DETERMINE WHETHER TO ACCEPT OR REJECT THE NULL
HYPOTHESIS
485 '****************************************
500 IF (OPER$="<") AND (TSTAT<-TALPHA) THEN
FLAG$="REJECT":RETURN
510 IF (OPER$=">") AND (TSTAT>TALPHA) THEN
FLAG$="REJECT":RETURN
520 IF (OPER$="<>" OR OPER$="><") AND ((TSTAT<-TALPHA) OR
(TSTAT>TALPHA)) THEN FLAG$="REJECT":RETURN
530 FLAG$="ACCEPT"
540 RETURN
585 '****************************************
590 'OUTPUT ROUTINE
595 '****************************************
600 CLS
605 PRINT"STEP 6. BASED OF THE RELATIONSHIP BETWEEN THE
COMPUTED TEST STATISTIC AND THE T-ALPHA VALUE OBTAINED FROM
THE CHART DETERMINE IF YOU CAN REJECT THE NULL HYPOTHESIS."
610 PRINT"REMEMBER, T-ALPHA= + OR -";TAFLHA
615 PRINT"TEST STAT=";TSTAT
620 INPUT "R-TO REJECT OR A-FOR ACCEPT";R$:CLS
625 IF R$<>"R" AND R$<>"A" AND R$<>"r" AND R$<>"a" THEN GOTO
```

```
610
630 IF (R$="R" OR R$="r") AND FLAG$="REJECT" THEN PRINT
"CORRECT":PRINT"WE SHOULD REJECT THE NULL HYP.":RETURN
635 IF (R$="A" OR R$="a") AND FLAG$="ACCEPT" THEN PRINT
"CORRECT":PRINT"WE CANNOT REJECT THE NULL HYP.":RETURN
640 PRINT"WRONG!"
645 PRINT"THESE VALUES SUGGEST WE SHOULD ";FLAG$
650 PRINT"BETTER LUCK NEXT TIME":RETURN
1080 '****************************************
1085 'CONFIDENCE INTERVAL ABOUT THE MEAN
1090 'CI DRIVER
1095 '****************************************
1100 CLEAR:GOSUB 1200
1102 GOSUB 20000
1105 GOSUB 1300
1110 GOSUB 1500
1120 GOSUB 1600
1125 PRINT
1130 INPUT "SELECT: 1-MORE C.I.; 2-MENU; 3-EXIT";Q
1140 ON Q GOTO 1100,100,145
1145 '****************************************
1150 'INTRODUCTION
1155 '****************************************
1200 CLS:PRINT"***CONFIDENCE INTERVAL ABOUT THE MEAN**"
1210 PRINT"            SOLUTION STEPS
1.DETERMINE SAMPLE SIZE(N) AND DEGREES    OF FREEDOM(N-1)
                  2.DETERMINE THE DEGREE OF CONFIDENCE(C)
REQUIRED IN THE PROBLEM (usually        stated as a %)"
1220 INPUT"PUSH ENTER TO CONTINUE";Q
1225 CLS
1230 PRINT "3.CONVERT C TO AN ALPHA VALUE [(1-C)/2]"
1240 PRINT "4.OBTAIN PROPER T-(ALPHA)-VALUE FROM THE  TABLE
(be sure to use th right     degree of freedom)
            5.COMPUTE FROM THE SAMPLE (if not given)  THE
SAMPLE STD. DEV. AND SAMPLE MEAN"
1250 PRINT "6.COMPUTE [PLUG AND JUG]"
1255 INPUT "PUSH ENTER TO CONTINUE";Q
1260 CLS:RETURN
1280 '****************************************
1285 INPUT ROUTINE
1290 '****************************************
1300 PRINT"STEP 1.":INPUT "INPUT SAMPLE SIZE";SNUM
1305 IF ABS(INT(SNUM))<>SNUM THEN GOTO 1300
1310 INPUT "INPUT DEGREES OF FREEDOM";DFRD
1320 IF DFRD<>(SNUM-1) THEN PRINT"WRONG!, TRY AGAIN": GOTO
1310
1330 CLS:PRINT"STEP 2."
1340 INPUT "INPUT DEGREE OF CONFIDENCE REQUIRED (in decimal
form)";DCNF
1345 IF DCNF>1 OR DCNF<0 THEN PRINT"THE D.ofC. MUST BE IN
```

```
POSITIVE DECIMAL FORM!":GOTO 1340
1350 PRINT"STEP 3."
1360 INPUT "INPUT ALPHA VALUE {as calculated from the deg.
of conf.:(1-C)/2}";LSIG
1370 IF LSIG<>(1-DCNF)/2 THEN PRINT"WRONG!!, TRY AGAIN":GOTO
1360
1380 S2=LSIG
1390 CLS: PRINT"STEP 4."
1400 INPUT "INPUT PROPER T-ALPHA VALUE FROM TABLE"; TALPHA
1410 GOSUB 15000:REM COMPARE INPUT VALUE TO TABLE VALUE
1415 FOR I=1 TO 500:NEXT I
1420 CLS:PRINT"STEP 5."
1430 INPUT "INPUT SAMPLE STANDARD DEV.";STDV
1435 IF ABS(STDV)<>STDV THEN PRINT"STDV MUST BE
POSITIVE":GOTO 1430
1440 INPUT "INPUT SAMPLE MEAN";SMEAN
1450 CLS:PRINT"STEP 6."
1455 PRINT:PRINT"PLUG VALUES INTO THE FORMULA AND COMPUTE"
1460 INPUT "PUSH ENTER WHEN READY TO CONTINUE!";Q
1470 RETURN
1480 '****************************************
1485 'CALCULATE THE UPPPER AND LOWER LIMITS
1490 '****************************************
1500 CLS:
1505 MER=TALPHA*STDV/SQR(SNUM)
1510 LFT=SMEAN-MER
1520 RGHT=SMEAN+MER
1530 RETURN
1580 '****************************************
1585 'OUTPUT ROUTINE
1590 '****************************************
1600 CLS:PRINT"THE CORRECT LOWER LIMIT=";LFT
1630 PRINT"THE CORRECT UPPER LIMIT=";RGHT
1635 PRINT"CONFIDENCE INTERVAL ="
1640 PRINT LFT;"< TRUE MEAN <";RGHT
1642 PRINT"MAX ERROR =";MER
1645 RETURN
1670 '****************************************
1675 MAXIMUM ERROR OF AN ESTIMATE
1680 DRIVER ROUTINE
1685 '****************************************
2050 CLEAR:CLS
2060 GOSUB 20000
2070 GOSUB 2110
2080 GOSUB 2300
2090 GOSUB 2400
2100 INPUT "SELECT: 1-MORE EST. OF MAX.ERROR;
2-MENU;3-EXIT";Q
2105 ON Q GOTO 2050,100,145
2106 '****************************************
```

```
2107 'DATA INPUT ROUTINE
2108 '****************************************
2110 PRINT"**********MAX ERROR OF ESTIMATE*********"
2120 PRINT:PRINT"STEP 1.":PRINT
2130 INPUT "INPUT SAMPLE SIZE";SNUM
2140 CLS:PRINT"STEP 2."
2150 PRINT:INPUT "INPUT THE LEVEL OF SIGNIFICANCE
{1-PROBABILITY}";LSIG
2155 IF LSIG>1 OR LSIG<0 THEN PRINT"MUST BE IN DECIMAL
FORM!":GOTO 2150
2160 S2=LSIG/2
2170 CLS:PRINT"STEP 3."
2180 INPUT "DIVIDE THE L. OF SIG. VALUE BY TWO TO ACCOUNT
FOR BOTH TAILS AND INPUT THIS VALUE";A2
2190 IF INT(1000*S2)/1000<>INT(1000*A2)/1000 THEN
CLS:PRINT"YOU HAVE MADE AN ERROR IN YOUR DIVISION OR IN YOUR
SPECIFICATION OF THE L.O.S.- TRY AGAIN":GOTO 2150
2200 CLS:PRINT"STEP 4.":PRINT
2210 INPUT "INPUT THE Z-VALUE FOR THIS VALUE FROM THE
TABLES";TALPHA
2220 GOSUB 15000:REM COMPARE INPUT VALUE TO TABLE VALUE
2225 FOR I=1TO 500:NEXT I
2230 CLS:PRINT"STEP 5.":PRINT
2240 INPUT "INPUT THE STAN. DEV. OF THE SAMPLE";STDV
2245 IF ABS(STDV)<>STDV THEN GOTO 2240
2250 RETURN
2280 '****************************************
2285 'CALCULATE THE MAX ERROR
2290 '****************************************
2300 MER=TALPHA*STDV/SQR(SNUM)
2310 RETURN
2380 '****************************************
2385 'OUTPUT ROUTINE
2390 '****************************************
2400 CLS:PRINT:PRINT"MAX ERROR OF EST=";MER
2410 FOR I=1 TO 1200:NEXT I
2420 RETURN
3070 '****************************************
3075 'DETERMINATION OF SAMPLE SIZE
3080 'DRIVER ROUTINE
3085 '****************************************
3100 CLS:CLEAR
3110 GOSUB 20000
3120 GOSUB 3200
3130 GOSUB 3370
3140 PRINT:INPUT"SELECT: 1-MORE DET. OF SAMPLE SIZE;
2-MENU;3-EXIT";Q
3150 ON Q GOTO 3100,100,145
3160 END
3180 '****************************************
```

```
3185 'DATA INPUT ROUTINE
3190 '****************************************
3200 PRINT"******DETERMINATION OF SAMPLE
SIZE************WHEN GIVEN:1.LEVEL OF SIGNIFICANCE
     2.ALLOWABLE ERROR                       3.STANDARD
DEV."
3210 PRINT "STEP 1.":PRINT
3220 INPUT"INPUT THE LEVEL OF SIGNIFICANCE (1 -
probability)";LSIG
3225 IF LSIG>1 OR LSIG<0 THEN PRINT"MUST BE BETWEEN 0 AND
1!":GOTO 3220
3230 S2=LSIG/2
3240 CLS:PRINT"STEP 2.":PRINT
3250 INPUT"DIVIDE THE L. OF SIG. BY TWO TO ACCOUNT FOR BOTH
TAILS AND INPUT THIS VALUE";A2
3260 IF INT(1000*S2)/1000<>INT(1000*A2)/1000 THEN PRINT"YOU
HAVE MADE AN ERROR IN YOUR DIVISION OR IN YOUR L.O.S. INPUT.
TRY AGAIN":GOTO 3220
3270 CLS:PRINT"STEP 3.":PRINT
3280 PRINT"LOOK UP THE PROPER Z-VALUE FOR Z-SUB";S2
3285 INPUT"INPUT THE PROPER Z-VALUE(t-alpha value)";TALPHA
3290 SNUM=33:GOSUB 15000:REM COMPARE INPUT VALUE TO TABLE
VALUE
3300 FOR I=1TO 500:NEXT I
3310 CLS:PRINT"STEP 4.":PRINT
3320 INPUT "INPUT THE LEVEL OF ALLOWABLE ERROR";MER
3330 CLS:PRINT"STEP 5.":PRINT
3340 INPUT"INPUT THE STANDARD DEV.";STDV
3345 IF ABS(STDV)<>STDV THEN GOTO 3340
3350 INPUT"PLUG THESE VALUES INTO FORMULA-10 AND COMPUTE THE
SAMPLE SIZE. PUSH ENTER AND THE COMPUTER WILL ASSIST!";Q
3360 RETURN
3361 '****************************************
3362 'OUTPUT ROUTINE
3363 '****************************************
3370 SNUM=(TALPHA*STDV/MER)^2
3380 CLS:PRINT"REQUIRED SAMPLE SIZE =";SNUM
3390 RETURN
14080 '****************************************
14085 'COMMON TABLE LOOK-UP ROUTINE
14090 '****************************************
15000 IF S2=.1 THEN PTR%=1
15005 IF S2=.05 THEN PTR%=6
15010 IF S2=.025 THEN PTR%=11
15015 IF S2=.010 THEN PTR%=16
15020 IF S2=.005 THEN PTR%=21
15025 IF PTR%=0 THEN PRINT "ODD LEVEL OF SIGNIFICANCE: NO
ERROR CHECKING AVAILABLE":RETURN
15030 IF SNUM>31 THEN TNUM=31 ELSE TNUM=SNUM
15035 TVL$= MID$(TABLE$(TNUM-2),PTR%,5)
```

```
15040 TL=VAL(TVL$)
15045 IF TALPHA<>TL THEN PRINT "WRONG VALUE TRY AGAIN!!"
ELSE PRINT "CORRECT":RETURN
15050 INPUT "INPUT T-ALPHA VALUE (LAST CHANCE)";TALPHA
15055 CLS
15060 IF TL<>TALPHA THEN PRINT "WRONG AGAIN!! THE CORRECT
VALUE IS";TL:TALPHA=TL ELSE PRINT "RIGHT ON"
15065 RETURN
19080 '***************************************
19085 'TABLE DATA
19090 '***************************************
20000 DEFSNG A-Z
20020 ON ERROR GOTO 22000
20055 DIM TABLE$(29)
20060 FOR L=0 TO 29
20065   READ TABLE$(L)
20070 NEXT L
20071 DATA "3.0786.31412.7031.8263.65"
20072 DATA "1.8862.9204.3036.9659.925"
20073 DATA "1.6382.3533.1824.5415.841"
20074 DATA "1.5332.1322.7763.7474.604"
20075 DATA "1.4762.0152.5713.3654.032"
20076 DATA "1.4401.9432.4473.1433.707"
20077 DATA "1.4151.8952.3652.9983.499"
20078 DATA "1.3971.8602.3062.8963.355"
20079 DATA "1.3831.8332.2622.8213.250"
20080 DATA "1.3721.8122.2282.7643.169"
20081 DATA "1.3631.7962.2012.7183.106"
20082 DATA "1.3561.7822.1792.6813.055"
20083 DATA "1.3501.7712.1602.6503.012"
20084 DATA "1.3451.7612.1452.6242.977"
20085 DATA "1.3411.7532.1312.6022.947"
20086 DATA "1.3371.7462.1202.5832.921"
20087 DATA "1.3331.7402.1102.5672.898"
20088 DATA "1.3301.7342.1012.5522.878"
20089 DATA "1.3281.7292.0932.5392.861"
20090 DATA "1.3251.7252.0862.5282.845"
20091 DATA "1.3231.7212.0802.5182.831"
20092 DATA "1.3211.7172.0742.5082.819"
20093 DATA "1.3191.7142.0692.5002.807"
20094 DATA "1.3181.7112.0642.4922.797"
20095 DATA "1.3161.7082.0602.4852.787"
20096 DATA "1.3151.7062.0562.4792.779"
20097 DATA "1.3141.7032.0522.4732.771"
20098 DATA "1.3131.7012.0482.4672.763"
20099 DATA "1.3111.6992.0452.4622.756"
20100 DATA "1.2821.6451.9602.3262.576"
20105 RETURN
21080 '***************************************
21085 ERROR ROUTINE
```

```
21090 '**************************************************
22000 PRINT "ERROR";ERR;"IN LINE";ERL
22010 PRINT"CONSULT OPERATORS MANUAL":END
```

```
0 'TITLE: MODULE 6
1 'DATE: 30 APR 85
2 'PURPOSE: THIS PROGRAM IS DESIGNED TO ASSIST THE STUDENT
IN LEARNING STANDARD PROBLEM SOLVING TECHNIQUES WHICH CAN BE
APPLIED TO THE FOLLOWING TYPES OF STATISTICAL PROBLEMS:
3 'HYPOTHESIS TESTING CONCERNING THE STANDARD DEVIATION.
4 'DERIVING A CONFIDENCE INTERVAL ABOUT THE STANDARD
DEVIATION
5 '
6 'A SEPARATE GROUP OF SUBROUTINES IS DEFINED FOR EACH
APPLICATION.
7 '
8 'SIGNIFICANT SUBROUTINES WHICH ARE JOINTLY USED BY BOTH
APPLICATIONS INCLUDE:
9 'A DATA TABLE ROUTINE WHICH PROVIDES CHI-SQUARE AND
STANDARD Z VALUES; AND TWO ROUTINES WHICH COMPARE A
STUDENT'S INPUT FOR CHI-SQUARE OR Z VALUES TO THE DATA
TABLE.
10 '
11 'INPUT VARIABLES:
12 'DCNF=DEGREE OF CONFIDENCE
13 'DFRD=DEGREES OF FREEDOM
14 'CHOICE$=DATA TABLE SELECTION
15 'LSIG=LEVEL OF SIGNIFICANCE
16 'NHYP=VALUE OF THE NULL HYPOTHESIS
17 'OPER$=OPERATOR OF THE ALTERNATE HYPOTHESIS
18 'R$=ACCEPT/REJECT CRITERIA(student)
19 'SNUM=SAMPLE SIZE
20 'STDV=STANDARD DEVIATION
21 'X1ALPHA:X2ALPHA = CHI-SQUARE VALUES
22 'ZALPHA=STANDARD Z VALUE
23 '
24 'OUTPUT VARIABLES
25 'FLAG$=ACCEPT/REJECT CRITERIA (computer)
26 'LFT=LOWER CI LIMIT
27 'RGT=UPPER CI LIMIT
28 'TL=CHI-SQUARE OR Z TABLE VALUE
29 'TSTAT=TEST STATISTIC
30 '
31 'INTERNAL VARIABLES
32 'F2LAG=USED AS A SWITCH
33 'PTR%=TABLE POINTER
34 'Q$:Q%:R%:S2 = INTERMEDIATE VALUES
35 'TABLE$=CHI-SQUARE TABLE
36 'TVL%=INTERMEDIATE VALUE
37 'ZZ!=STANDARD Z TABLE
38 '
39 '
85 '****************************************
90 'TOTAL PROGRAM MENU SELECTION
```

```
95 '****************************************
100 CLS:PRINTTAB(18);"MENU"
110 PRINT"1-CONFIDENCE INTERVAL ABOUT ST.DV."
120 PRINT"2-HYPOTHESIS TESTING OF ST.DV."
130 PRINT"3-EXIT"
135 PRINT:PRINT
140 INPUT "SELECTION";Q
150 ON Q GOTO 500,4850,160
160 MENU
475 '========================================
480 '========================================
485 '****************************************
490 CONFIDENCE INTERVAL DRIVER
495 '****************************************
500 CLEAR:GOSUB 20000
503 F2LAG$="INT"
505 GOSUB 1000
510 IF DFRD<30 THEN GOSUB 2000 ELSE GOSUB 3000
520 GOSUB 4000
530 PRINT:INPUT "SELECT: 1-MENU; 2-MORE CONF.INTERVALS;
3-EXIT";Q
540 ON Q GOTO 100,500,560
560 MENU
980 '****************************************
985 'BASIC DATA INPUT ROUTINE
990 '****************************************
1000 CLS:PRINT"***CONFIDENCE INTERVAL ABOUT THE STDV**"
1010 PRINTTAB(12)"SOLUTION STEPS"
1020 PRINT:PRINT"STEP 1."
1030 PRINT"DETERMINE THE SAMPLE SIZE AND THE STANDARD
DEV.":PRINT
1040 INPUT "INPUT SAMPLE SIZE, AND STD.DEV";SNUM,STDV
1044 IF ABS(INT(SNUM))<>SNUM THEN PRINT"SAMPLE SIZE MUST BE
A POSITIVE INTERER; TRY AGAIN!":GOTO 1040
1045 IF ABS(STDV)<>STDV THEN PRINT"ST.DEV. IS ALWAYS
POSITIVE; ENTER BOTH VALUES AGAIN!":GOTO 1040
1050 CLS:PRINT"STEP 2."
1060 PRINT"COMPUTE THE DEGREES OF FREEDOM FROM THE
SAM.SIZE.":PRINT
1070 INPUT"INPUT THE DEG. OF FRE.";DFRD
1080 IF DFRD<>(SNUM-1) THEN PRINT"WRONG! TRY AGAIN (REMEMBER
DF=S.SIZE-1  OR";SNUM;"-1)":GOTO 1070
1090 CLS:PRINT"STEP 3."
1100 PRINT"DETERMINE THE LEVEL OF CONFIDENCE
REQUIRED.":PRINT
1110 INPUT "INPUT THE L. OF CONFIDENCE (C) IN DECIMAL
FORM";DCNF
1120 IF DCNF>1 OR DCNF<0 THEN PRINT"THE INPUT SHOULD BE IN
DECIMAL NOT %!! TRY AGAIN":GOTO 1110
1130 CLS:PRINT"STEP 4."
```

```
1140 PRINT"CONVERT C TO AN ALPHA VALUE (1-C).":PRINT
1150 INPUT "INPUT ALPHA VALUE";LSIG
1160 IF LSIG<>(1-DCNF) THEN PRINT "WRONG!! TRY AGAIN":GOTO
1140
1170 CLS:PRINT"STEP 5."
1180 PRINT"IF THE DEGREES OF FREEDOM IS > 30 THEN USE THE
'Z' TABLE TO FIND A VALUE FOR Z(ALPHA/2)."
1190 PRINT"IF THE D.O.F IS < 30 THEN YOU MUST DETERMINE TWO
VALUES FROM THE CHI-SQUARE 'X' TABLE: X(ALPHA/2) and
X(1-ALPHA/2)"
1200 INPUT "WHICH TABLE DO YOU NEED? Z or X";Q$
1210 IF (DFRD>30 AND Q$="Z") OR (DFRD<=30 AND Q$="X") THEN
PRINT"CORRECT" ELSE PRINT"WRONG"
1220 FOR I=1 TO 300:NEXT I
1230 RETURN
1980 '****************************************
1985 'CHI-SQUARE VALUE INPUT / C.I. CALCULATION (for N<30)
1990 '****************************************
2000 IF F2LAG$="INT" THEN CLS:PRINT"STEP 5.
(CONTINUED)":PRINT"SINCE THE DEGREES OF FREEDOM =";DFRD;"YOU
NEED TO DETERMINE X(ALPHA/2) AND X(1-ALPHA/2)."
2010 PRINT"LET'S START WITH X(ALPHA/2)
2020 PRINT"REMEMBER ALPHA/2 =";LSIG/2
2030 INPUT "INPUT X(ALPHA/2) FROM CHI-SQUARE TABLE";X1ALPHA
2040 XALPHA=X1ALPHA:S2=LSIG/2
2050 GOSUB 15000
2060 FOR I=1 TO 500:NEXT I
2070 X1ALPHA=XALPHA
2080 CLS:PRINT"NEXT INPUT THE VALUE FOR X(1-ALPHA/2):
REMEMBER 1-ALPHA/2=";1-LSIG/2
2090 INPUT "INPUT X(1-ALPHA/2) FROM THE TABLE";X2ALPHA
2100 XALPHA=X2ALPHA:S2=1-LSIG/2
2110 GOSUB 15000:REM ***COMPARE INPUT VALUE TO TABLE
VALUE***
2115 FOR I=1 TO 500:NEXT I
2120 X2ALPHA=XALPHA
2130 CLS
2135 IF F2LAG$="INT" THEN PRINT"STEP 6." ELSE PRINT"STEP 7."
2140 PRINT"PLUG THE VALUES INTO FORMULA-13 OR FORMULA-14 AND
COMPUTE.":PRINT
2150 INPUT "PUSH ENTER AND THE COMPUTER WILL ASSIST";Q
2160 LFT=SQR((SNUM-1)*(STDV^2)/X1ALPHA)
2170 RGT=SQR((SNUM-1)*(STDV^2)/X2ALPHA)
2190 RETURN
2985 '****************************************
2990 'STANDARD Z-VALUE INPUT / C.I. CALCULATION (for N>30)
2995 '****************************************
3000 CLS:PRINT"STEP 5.(continued)"
3010 PRINT"SINCE THE DEGREES OF FREEDOM =";DFRD;"YOU ONLY
NEED TO LOOK UP Z(ALPHA/2) FROM THE Z-TABLE."
```

```
3020 PRINT"REMEMBER ALPHA/2=";LSIG/2:PRINT
3030 INPUT "INPUT Z(ALPHA/2) FROM THE Z TABLE";ZALPHA
3040 S2=LSIG/2
3050 GOSUB 15500:REM ***COMPARE INPUT VALUE TO TABLE
VALUE***
3060 FOR I= 1 TO 500:NEXT I
3070 CLS:PRINT"STEP 6."
3080 PRINT"PLUG THE VALUES INTO FORMULA-12 AND
COMPUTE.":PRINT
3090 INPUT "PUSH ENTER AND THE COMPUTER WILL ASSIST. YOU!";Q
3100 LFT=(STDV/(1+(ZALPHA/SQR(2*SNUM))))
3110 RGT=(STDV/(1-(ZALPHA/SQR(2*SNUM))))
3130 RETURN
3980 '****************************************
3985 'OUTPUT ROUTINE
3990 '****************************************
4000 CLS:PRINT"GIVEN A STANDARD DEV. OF";STDV;"AND A SAMPLE
SIZE OF";SNUM
4010 PRINT"THE LOWER LIMIT=";LFT
4020 PRINT"THE UPPER LIMIT=";RGT:PRINT
4030 PRINTLFT;"< TRUE ST.DV.<";RGT
4040 FOR I=1 TO 1000:NEXT I
4050 RETURN
4825 '========================================
4830 '========================================
4835 '****************************************
4840 'HYPOTHESIS TESTING DRIVER
4845 '****************************************
4850 CLEAR:GOSUB 20000
4860 GOSUB 9000
4870 IF DFRD<30 THEN GOSUB 5000 ELSE GOSUB 6000
4875 GOSUB 5500
4880 GOSUB 7000
4890 GOSUB 8000
4900 PRINT:INPUT"SELECT: 1-MENU; 2-MORE HYP. TESTING;
3-EXIT";Q
4910 ON Q GOTO 100,4850,160
4920 END
4980 '*****************************************
4985 'CHI-SQUARE VALUE INPUT AND TEST STAT. CALCULATION
4990 '*****************************************
5000 CLS:PRINT"STEP 6.(continued)"
5005 IF OPER$="<>" OR OPER$="><" THEN PRINT"SINCE DEGREES OF
FREEDOM=";DFRD;"AND THE SIGN OF THE ALT.HYP. IS <>,YOU WILL
HAVE TO LOOK UP TWO VALUES IN THE CHI-SQUARE
TABLE":PRINT"X(alpha/2) and X(1-alpha/2)":PRINT
5010 IF OPER$="<>" OR OPER$="><" THEN INPUT"PUSH ENTER TO
CONTINUE";Q:CLS:GOSUB 2000:REM ***GO TO CHI-SQUARE
INPUT/CHECK ROUTINE***
5015 IF OPER$="<>" OR OPER$="><" THEN GOTO 5110
```

```
5020 PRINT"SINCE THE DEGREES OF FREEDOM =";DFRD;"AND THE
SIGN OF THE ALT.HYP. IS ";OPER$;" YOU WILL ONLY BE REQUIRED
TO LOOK UP ONE VALUE IN THE CHI-SQUARE (X) TABLE!!":PRINT
5025 INPUT"PUSH ENTER TO CONTINUE";Q:CLS
5030 IF OPER$="<" THEN PRINT"IN THIS CASE (<) FIND THE
X(1-alpha) FROM THE CHI-SQUARE TABLE":S2=1-LSIG ELSE
PRINT"IN THIS CASE (>) FIND THE X(alpha) FROM THE CHI-SQUARE
TABLE":S2=LSIG
5040 PRINT"INPUT THE VALUE FOR";DFRD;"DEGREES OF FREEDOM,
FOR X(";S2;")"
5050 INPUT XALPHA
5060 GOSUB 15000:REM ***COMPARE INPUT VALUE TO TABLE
VALUE***
5065 X1ALPHA=XALPHA:X2ALPHA=XALPHA
5070 FOR I=1 TO 300:NEXT I
5080 CLS:PRINT"STEP 7."
5090 PRINT"THE NEXT STEP IS TO COMPUTE THE TEST STATISTIC.
(Formula-13)":PRINT:PRINT
5100 INPUT "PUSH ENTER AND THE COMPUTER WILL ASSIST!";Q
5110 CLS:TSTAT=((SNUM-1)*(STDV^2)/NHYP^2)
5120 RETURN
5480 '******************************************
5485 'TEST STATISTIC PRINT ROUTINE
5490 '******************************************
5500 PRINT"USING THE FOLLOWING VALUES:"
5510 PRINT"NULL HYP=";NHYP
5520 PRINT"SAMPLE SIZE=";SNUM
5530 PRINT"STD.DEV.=";STDV
5540 PRINT:PRINT"THE TEST STATISTIC WILL BE=";TSTAT
5550 FOR I=1 TO 1700:NEXT I
5560 RETURN
5585 '******************************************
5590 'Z-VALUE INPUT AND TEST STAT. CALCULATION
5595 '******************************************
6000 CLS
6010 IF OPER$="<>" OR OPER$="><" THEN PRINT"SINCE THE
DEGREES OF FREEDOM =";DFRD;"AND THE OPERATOR OF THE ALT.HYP.
IS <>, YOU ONLY NEED TO LOOK UP Z(ALPHA/2) FROM THE
Z-TABLE.":S2=LSIG/2
6015 IF OPER$<>"<>" AND OPER$<>"><" THEN PRINT "SINCE THE
DEGREES OF FREEDOM =";DFRD;"AND THE OPERATOR OF THE ALT.HYP.
IS ";OPER$;" YOU ONLY NEED TO LOOK UP Z(ALPHA) IN THE
Z-TABLE":S2=LSIG
6020 PRINT"REMEMBER THIS MEANS THE ALPHA  LEVEL YOU NEED
IS";S2:PRINT
6030 INPUT "INPUT VALUE FROM THE Z TABLE";ZALPHA
6050 GOSUB 15500:REM ***COMPARE INPUT VALUE TO TABLE
VALUE***
6060 FOR I= 1 TO 500:NEXT I
6070 CLS:PRINT"STEP 7."
```

```
6080 PRINT"THE NEXT STEP IS TO COMPUTE THE TEST
STATISTIC.":PRINT
6090 INPUT"PUSH ENTER AND THE COMPUTER WILL ASSIST!";Q
6100 CLS:TSTAT=((STDV-NHYP)/(NHYP/SQR(2*SNUM)))
6110 X2ALPHA=-ZALPHA:X1ALPHA=ZALPHA
6120 RETURN
6980 '****************************************
6985 'DETERMINE: ACCEPT/REJECT OF NULL HYPOTHESIS
6990 '****************************************
7000 IF (OPER$="<") AND (TSTAT<X2ALPHA) THEN
FLAG$="REJECT":RETURN
7010 IF (OPER$=">") AND (TSTAT>X1ALPHA) THEN
FLAG$="REJECT":RETURN
7020 IF (OPER$="<>" OR OPER$="><") AND ((TSTAT<X2ALPHA) OR
(TSTAT>X1ALPHA)) THEN FLAG$="REJECT":RETURN
7030 FLAG$="ACCEPT"
7040 RETURN
7985 '****************************************
7990 'OUTPUT ROUTINE
7995 '****************************************
8000 CLS:PRINT"STEP 8."
8010 PRINT"BASED ON THE RELATIONSHIP BETWEEN THE COMPUTED
TEST STATISTIC AND THE VALUE(S) OBTAINED FROM THE TABLE,
DETERMINE IF YOU CAN REJECT THE NULL HYPOTHESIS."
8020 INPUT "PUSH ENTER TO CONTINUE";Q
8030 CLS:PRINT"REMEMBER:"
8040 PRINT"THE TEST STAT.=";TSTAT;"
8050 IF OPER$<>"<>" AND OPER$<>"><" THEN PRINT"THE TABLE
VALUE =";X1ALPHA ELSE PRINT"THE TWO TABLE VALUES
=";X2ALPHA;"AND";X1ALPHA
8060 INPUT"INPUT:R-TO REJECT OR A-TO ACCEPT";R$
8070 IF R$<>"R" AND R$<>"A" THEN PRINT"YOU MUST ENTER
CAPITAL R OR A!":GOTO 8060
8080 CLS
8090 IF R$="R" AND FLAG$="REJECT" THEN
PRINT"CORRECT":PRINT"WE SHOULD REJECT THE NULL
HYPOTHESIS":RETURN
8100 IF R$="A" AND FLAG$="ACCEPT" THEN
PRINT"CORRECT":PRINT"WE CANNOT REJECT THE NULL
HYPOTHESIS":RETURN
8110 PRINT"WRONG"
8120 PRINT"THESE VALUES SUGGEST WE SHOULD ";FLAG$;" THE NULL
HYPOTHESIS"
8130 PRINT"BETTER LUCK NEXT TIME"
8140 RETURN
8985 '****************************************
8990 'HYPOTHESIS TEST DATA INPUT ROUTINE
8995 '****************************************
9000 CLS:PRINT"***HYPOTHESIS TESTING FOR THE ST. DEV.**"
9010 PRINT"*************SOLUTION STEPS***********"
```

```
9020 PRINT"STEP 1."
9030 PRINT"FORMULATE THE NULL HYPOTHESIS.":PRINT
9040 INPUT "INPUT THE VALUE OF THE NULL HYP.";NHYP
9050 CLS:PRINT"STEP 2."
9060 PRINT"FORMULATE THE ALTERNATIVE HYPOTHESIS.":PRINT
9070 INPUT"INPUT THE OPERATOR OF THE ALT. HYP.: < (less
than), > (greater than), OR <> (not equal to)";OPER‡
9075 IF OPER‡<>"<" AND OPER‡<>">" AND OPER‡<>"<>" AND
OPER‡<>"><" THEN GOTO 9070
9080 CLS:PRINT"BASED ON THE OPERATOR OF THE ALT. HYP. IS
THIS A 1 OR 2 TAILED TEST?"
9090 INPUT Q%:CLS
9100 IF Q%<>1 AND Q%<>2 THEN GOTO 9080
9110 IF OPER‡="<>" OR OPER‡="><" THEN R%=2 ELSE R%=1
9120 IF Q%=R% THEN PRINT"CORRECT" ELSE PRINT"WRONG"
9130 IF OPER‡="<>" OR OPER‡="><" THEN PRINT"<> INDICATES A
TWO TAILED TEST" ELSE PRINT"< OR > INDICATE A ONE TAILED
TEST"
9135 FOR I=1 TO 500:NEXT I
9140 CLS:PRINT"STEP 3."
9150 PRINT"DETERMINE THE LEVEL OF SIGNIFICANCE (alpha) AS
STATED IN THE PROBLEM.":PRINT
9160 INPUT "INPUT LEVEL OF SIG.";LSIG
9165 IF LSIG>1 OR LSIG<0 THEN PRINT"L. of S. MUST BE A
DECIMAL VALUE BETWEEN 0 AND 1.":GOTO9160
9170 CLS:PRINT"STEP 4."
9180 PRINT"EXTRACT SAMPLE SIZE AND STANDARD DEV. FROM THE
PROBLEM.":PRINT
9190 INPUT "INPUT SAMPLE SIZE, STD.DEV.";SNUM,STDV
9194 IF ABS(INT(SNUM))<>SNUM THEN PRINT"SAM.SZ. MUST BE A
POSITIVE INTEGER; TRY AGAIN!":GOTO 9190
9195 IF ABS(STDV)<>STDV THEN PRINT"ST.DV. MUST BE
POSITIVE;ENTER BOTH VALUES AGAIN!":GOTO 9190
9200 CLS:PRINT"STEP 5."
9210 PRINT"DETERMINE THE DEGREES OF FREEDOM FROM THE SAMPLE
SIZE (SAM.SZ.-1 OR";SNUM;"-1).":PRINT
9220 INPUT "INPUT THE DEGREES OF FREEDOM";DFRD
9230 IF DFRD <>SNUM-1 THEN PRINT"WRONG!TRY AGAIN":GOTO 9210
9240 CLS:PRINT"STEP 6"
9250 PRINT"IF THE DEGREES OF FREEDOM ARE > THAN 30 THEN YOU
ONLY NEED TO FIND ONE VALUE FROM THE Z-TABLE. IF THE D.O.F.
ARE < 30 THEN YOU NEED TO FIND THE VALUE(S) FROM THE
CHI-SQUARE (X) TABLE."
9260 INPUT "WHICH TABLE DO YOU NEED-Z OR X";CHOICE‡
9270 IF CHOICE‡ <>"Z" AND CHOICE‡<>"X" THEN GOTO 9260
9280 IF (DFRD>30 AND CHOICE‡="Z") OR (DFRD<=30 AND
CHOICE‡="X") THEN PRINT"CORRECT" ELSE PRINT"WRONG"
9290 RETURN
14985 '****************************************
14990 'CHI-SQUARE VALUE LOOKUP ROUTINE
```

```
14995 '****************************************
15000 FOR I=0 TO 7
15010   IF S2=ZZ!(I,0) THEN PTR%=ZZ!(I,1)
15020 NEXT I
15030 IF PTR%=0 THEN PRINT"ODD LEVEL OF SIG.:NO ERROR
CHECKING AVAILABLE!!":RETURN
15040 TVL$=MID$(TABLE$(SNUM-2),PTR%,6)
15050 TL=VAL(TVL$)*.01
15060 IF XALPHA<>TL THEN PRINT"WRONG VALUE! TRY AGAIN" ELSE
PRINT"CORRECT":RETURN
15070 INPUT "TABLE VALUE AGAIN";XALPHA
15080 IF XALPHA<>TL THEN PRINT"WRONG AGAIN!!THE CORRECT
VALUE IS";TL:XALPHA=TL ELSE PRINT"RIGHT ON"
15085 RETURN
15485 '****************************************
15490 'STANDARD Z-VALUE LOOKUP ROUTINE
15495 '****************************************
15500 IF S2=.1   THEN TL=1.282
15510 IF S2=.05  THEN TL=1.645
15520 IF S2=.025 THEN TL=1.96
15530 IF S2=.010 THEN TL=2.326
15540 IF S2=.005 THEN TL=2.576
15550 IF TL<1 THEN PRINT"NO ERROR CHECKING AVAILABLE":RETURN
15560 IF ZALPHA<>TL THEN PRINT"WRONG VALUE" ELSE
PRINT"CORRECT":RETURN
15570 PRINT"THE CORRECT VALUE IS";TL
15580 ZALPHA=TL
15590 RETURN
19980 '****************************************
19985 'CHI-SQUARE / Z-VALUE TABLE DATA
19990 '****************************************
20000 DEFSNG A-Z
20020 ON ERROR GOTO 22000
20030 DIM TABLE$(29)
20040 FOR I=0 TO 29
20045   READ TABLE$(I)
20050 NEXT I
20051 DATA
".003930.01570.09820.3930384.10502.40663.50787.90"
20052 DATA
"1.00002.01005.060010.300599.10737.80921.001059.7"
20053 DATA
"7.170011.50021.60035.200781.50934.801134.51283.8"
20054 DATA
"20.70029.70048.40071.100948.801114.31327.71486.0"
20055 DATA
"41.20055.40083.100114.501107.01283.21506.61675.0"
20056 DATA
"67.60087.200123.70163.501259.21444.91681.21854.8"
20057 DATA
```

```
"98.900123.90169.00216.701406.71601.31847.52027.8"
20058 DATA
"134.40164.60218.00273.301550.71753.52009.02195.5"
20059 DATA
"173.50208.80270.00332.501691.91902.32166.62358.9"
20060 DATA
"215.60255.80324.70394.001830.72048.32320.92518.8"
20061 DATA
"260.30305.30381.60457.501967.52192.02472.52675.7"
20062 DATA
"307.40357.10440.40522.602102.62333.72621.72830.0"
20063 DATA
"356.50410.70500.90589.202236.22473.62768.82981.9"
20064 DATA
"407.50466.00562.90657.102368.52611.92914.13131.9"
20065 DATA
"460.10522.90626.20726.102499.62748.83057.83280.1"
20066 DATA
"514.20581.20690.80796.202629.62884.53200.03426.7"
20067 DATA
"569.70640.80756.40867.202758.73019.13340.93571.8"
20068 DATA
"626.50701.50823.10939.002886.93152.63480.53715.6"
20069 DATA
"684.40763.30890.701011.73014.43285.23619.13858.2"
20070 DATA
"743.40826.00959.101085.13441.03417.03756.63999.7"
20071 DATA
"803.40889.701028.31159.13267.13547.93893.24140.1"
20072 DATA
"864.30954.201098.21233.83392.43678.14028.94279.6"
20073 DATA
"926.001019.61168.91309.13517.23807.64163.84418.1"
20074 DATA
"988.601085.61240.11384.83641.53936.44298.04555.8"
20075 DATA
"1052.01152.41312.01461.13765.24064.64431.44692.8"
20076 DATA
"1116.01219.81384.41537.93888.54192.34564.24829.0"
20077 DATA
"1180.81287.91457.31615.14011.34319.44696.34964.5"
20078 DATA
"1246.11356.51530.81692.84133.74446.14827.85099.3"
20079 DATA
"1312.11425.61604.71770.84255.74572.24958.85233.6"
20080 DATA
"1378.71495.31679.11849.34377.34697.95089.25367.2"
20100 DIM ZZ!(7,1)
20110 FOR I=0 TO 7
20115   FOR J=0 TO 1
```

```
20120     READ ZZ!(I,J)
20125   NEXT J
20230 NEXT I
20240 DATA
0.995,1,.99,7,.975,13,.95,19,.05,25,.025,31,.01,37,.005,43
20250 RETURN
21980 '****************************************
21985 'ERROR TRAP ROUTINE
21990 '****************************************
22000 PRINT"ERROR";ERR;"IN LINE";ERL
22020 PRINT"CONSULT OPERATORS MANUAL"
22030 END
```

```
0 'TITLE: MODULE 7
1 'DATE: 16 MAY 85
2 'PURPOSE: THIS PROGRAM IS DESIGNED TO ASSIST THE STUDENT
IN LEARNING STANDARD PROBLEM SOLVING TECHNIQUES WHICH CAN BE
APPLIED TO SOVING PROBLEMS RELATED TO LINEAR REGRESSION AND
ANALYSIS.
3 'TYPES OF PROBLEMS ADDRESSED INCLUDE THE FOLLOWING:
4 'HYPOTHESIS TESTING ON EITHER ALPHA OR BETA.
5 'FINDING A CONFIDENCE INTERVAL ABOUT ALPHA OR BETA
6 'CALCULATING A CONFIDENCE INTERVAL FOR THE MEAN OF Y AT Xo
7 'DETERMINING THE LIMITS OF PREDICTION
8 'OBTAINING ESTIMATES OF Y WHEN GIVEN X
9 '
10 'A SEPARATE GROUP OF SUBROUTINES IS DEFINED FOR EACH OF
THE ABOVE APPLICATIONS
11 '
12 'SIGNIFICANT SUBROUTINES WHICH ARE USED JOINTLY BY ALL
APPLICATIONS INCLUDE:
13 'A DATA TABLE ROUTINE WHICH PROVIDES  T-ALPHA VALUES; AND
A ROUTINE WHICH CHECKS A STUDENT'S INPUT FOR T-ALPHA VALUES
TO THE DATA TABLE.
14 '
15 'INPUT VARIABLES:
16 'DCNF=DEGREES OF CONFIDENCE
17 'LSIG=LEVEL OF SIGNIFICANCE
18 'OPER$=OPERATOR (used in HYP.TESTING)
19 'NHYP=VALUE OF THE NULL HYPOTHESIS
20 'Q%=STUDENT GUESS RELATING TO ONE AND TWO TAILED TESTS
21 'TALPHA=T-ALPHA VALUE (can be input by the student or
computer generated!)
22 'X=X VALUE IN THE INPUT DATA SET
23 'XNAUGHT=VALUE FOR Xo
24 'Y=VALUE IN THE INPUT DATA SET
25 '
26 'OUTPUT VARIABLES
27 'A=Y INTERCEPT IN LEAST SQUARES LINE
28 'ALOWER/AUPPER=LOWER AND UPPER C.I. LIMITS ABOUT ALPHA
29 'ATSTAT=TEST STATISTIC FOR HYP. TESTING OF ALPHA
30 'B=SLOPE IN LEAST SQUARES LINE
31 'BLOWER/BUPPER=LOWER AND UPPER C.I. LIMITS ABOUT BETA
32 'BTSTAT=TEST STATISTIC FOR HYP. TESTING OF BETA
33 'LLOWER/LUPPER=LOWER AND UPPER VALUES FOR LIMITS OF
PREDICTION
34 'TL=COMPUTER GENERATED VALUE FOR T-ALPHA
35 'SE=STANDARD ERROR
36 'TSTAT=GENERIC TEST STATISTIC
37 'ULOWER/UUPPER=LOWER AND UPPER C.I. VALUES ABOUT THE MEAN
OF Yo
38 'XSUM=SUM OF ALL X FROM THE INPUT DATA SET
39 'XYSUM=SUM OF X*Y FROM THE INPUT DATA SET
```

```
40 'X2SUM=SUM OF ALL X-SQUARED FROM THE INPUT DATA SET
41 'YSUM=SUM OF ALL Y FROM THE INPUT DATA SET
42 'Y2SUM=SUM OF ALL Y-SQUARED FROM THE INPUT DATA SET
43 '
44 'INTERNAL VARIABLES
45 'ACNF=INTERMEDIATE VALUE IN C.I. ABOUT ALPHA
46 'BCNF=INTERMEDIATE VALUE IN C.I. ABOUT BETA
47 'FLAG$=ACCEPT/REJECT CRITEREA (on Null Hyp.)
48 'GALPHA/GBETA=VALUE OF NULL HYPOTHESIS (clones of NHYP)
49 'I=LOOP INDEX
50 'LCNF=INTERMEDIATE VALUE FOR LIMITS OF PREDICTION
CALCULATION
51 'MEAN=AVG. OF ALL X VALUES FROM THE INPUT DATA SET
52 'N=TOTAL NUMBER OF INPUT DATA PAIRS
53 'PTR%=POINTER (used in table look-up)
54 'Q=USED IN PATH SELECTION
55 'Q5=X VALUE
56 'R%=DIVISOR OF LSIG (always = 1 or 2)
57 'SNUM=NUMBER OF INPUT DATA SETS (equivalent to N)
58 'SWITCH$=USED IN PATH SELECTION
59 'S2=NORMALIZED LSIG VALUE
60 'TABLE$=T-ALPHA TABLE VALUES
61 'TNUM=MODIFIED VALUE OF THE NUMER OF INPUT DATA SETS
62 'TVL$=STRING VALUE FOR T-ALPHA
63 'XMEAN2=AVG. OF ALL X-SQUARE FORM THE DATA SET
64 '
85 '****************************************
90 'INTRODUCTION TO LINEAR REGRESSION
95 '****************************************
100 CLS:PRINT"*****LINEAR REGRESSION + ANALYSIS*****"
110 PRINT "LINEAR REGRESSION REQUIRES THE USE OF A SERIES OF
COMPLEX EQUATIONS TO EXTRACT  USABLE INFORMATION FROM A DATA
SET. ";
120 PRINT"A   SIMPLE TABLE OF VALUES IS THE BEST      METHOD
TO OBTAIN THE INFORMATION NEEDED TO PLUG INTO THE
EQUATIONS!"
130 INPUT"PUSH ENTER TO CONTINUE";Q
140 CLS:PRINT"THE FIRST STEP IS TO SET UP YOUR TABLE  TO
OBTAIN VALUES FOR:"
150 PRINT"X-SUM","Y-SUM"
160 PRINT"X^2SUM","Y^2SUM"
170 PRINT"X*YSUM"
180 INPUT"PUSH ENTER AND THE COMPUTER WILL ASSIST YOU!";Q
190 GOSUB 20000:REM ***INITIALIZE TABLE DATA***
890 '****************************************
895 'BASIC DATA OUTPUT (INCLUDING THE LEAST-SQUARES LINE)
900  CLS
910 XSUM=0:X2SUM=0:YSUM=0:Y2SUM=0:XYSUM=0
1000 GOSUB 2000:REM ***INPUT DATA***
1005 IF N<4 THEN PRINT"YOU NEED MORE THAN TWO DATA
```

```
PAIRS!":END
1010 GOSUB 2750:REM ***CALCULATE VALUES FROM INPUT DATA***
1020 CLS:PRINT"X-SUM=";XSUM,"Y-SUM";YSUM
1030 PRINT"X^2SUM=";X2SUM
1040 PRINT"Y^2SUM=";Y2SUM
1050 PRINT"X*YSUM=";XYSUM
1060 PRINT"LEAST-SQUARES LINE:Y-hat=";A;"+";B;"X"
1070 PRINT"WHERE a=";A;"AND b=";B
1080 INPUT "PUSH ENTER TO CONTINUE";Q
1083 '****************************************
1084 'TOTAL PROGRAM MENU SELECTION
1085 '****************************************
1090 CLS:PRINT TAB(18);"MENU"
1100 PRINT"1-HYP.TEST"
1110 PRINT"2-CONF.INT.(for alfa or beta)"
1120 PRINT"3-CONF.INT.(mean of Y at Xo)"
1130 PRINT"4-LIMITS OF PRED."
1140 PRINT"5-ESTIMATES OF y WHEN GIVEN x"
1150 PRINT"6-EXIT"
1160 INPUT "SELECTION";Q
1170 ON Q GOTO 3000,4000,5100,6000,7000,1180
1175 END
1180 MENU
1980 '****************************************
1985 'DATA INPUT AND DISPLAY
1990 '****************************************
2000 N=1
2010 PRINT@0,"INPUT X";N;"(-999 when done)"
2020 INPUT X
2025 IF X=-999 THEN RETURN
2030 PRINT@40,"INPUT Y";N;
2040 INPUT Y
2050 CLS:PRINT@82,"X";
2060 PRINT@90,"Y";
2070 PRINT@202,"X^2";
2080 PRINT@215,"Y^2";
2090 PRINT@228,"X*Y";
2100 PRINT@120,X;
2110 PRINT@128,Y;
2120 PRINT@240,X^2;
2130 PRINT@254,Y^2;
2140 PRINT@268,X*Y;
2150 GOSUB 2500
2160 N=N+1
2170 GOTO 2010
2480 '****************************************
2485 'BASIC DATA COMPUTATIONS (INCLUDING STANDARD ERROR AND
LEAST-SQUARES LINE)
2490 '****************************************
2500 XSUM=XSUM+X
```

```
2510 X2SUM=X2SUM+X^2
2520 YSUM=YSUM+Y
2530 Y2SUM=Y2SUM+Y^2
2540 XYSUM=XYSUM+(X*Y)
2550 RETURN
2750 N=N-1:MEAN=XSUM/N:XMEAN2=MEAN^2:SNUM=N
2760 B=(N*XYSUM-XSUM*YSUM)/(N*X2SUM-XSUM^2)
2770 A=(YSUM-B*XSUM)/N
2780 SE=SQR((Y2SUM-A*YSUM-B*XYSUM)/(N-2))
2790 RETURN
2970 '=========================================
2975 '=========================================
2980 '*****************************************
2985 'HYPOTHESIS TESTING DRIVER
2990 '*****************************************
3000 CLS:PRINT"***********HYPOTHESIS TESTING**********"
3010 PRINT"WOULD YOU LIKE TO TEST alfa OR beta?"
3020 INPUT "A-FOR alpha:B-FOR beta";SWITCH$
3030 GOSUB 3100
3040 IF SWITCH$="A" THEN GOSUB 9000 ELSE GOSUB 9500
3050 GOSUB 3500
3060 GOSUB 3600
3070 PRINT:PRINT"SELECT: 1-MENU (same data); 2-MORE HYP.
TESTING; 3-TO ENTER NEW DATA SET; 4-EXIT";
3080 INPUT Q
3090 ON Q GOTO 1090,3000,900,1180
3095 END
3096 '*****************************************
3097 'DATA INPUT ROUTINE
3098 '*****************************************
3100 CLS:PRINT"*****************SOLUTION STEPS***********"
3105 PRINT"STEP 1. FORMULATE THE NULL HYPOTHESIS"
3110 PRINT:INPUT "INPUT VALUE OF THE NULL HYP.";NHYP
3120 GALPHA=NHYP:GBETA=NHYP
3125 CLS:PRINT "STEP 2. FORMULATE THE ALTERNATE HYPOTHESIS"
3130 PRINT:INPUT "IS THE OPERATOR OF THE ALT. HYP. < (less
than), > (greater than) OR >< (not     equal to)";OPER$
3135 IF ((OPER$<>"<") AND (OPER$<>">") AND (OPER$<>"><"))
THEN GOTO 3130
3140 IF OPER$="><" THEN R%=2 ELSE R%=1
3145 CLS:PRINT "BASED ON THE OPERATOR OF THE ALT. HYP.  IS
THIS A 1 OR 2 TAILED TEST?"
3150 INPUT Q%:CLS
3155 IF Q%<>1 AND Q%<>2 THEN GOTO 3145
3160 IF Q%=R% THEN PRINT "CORRECT" ELSE PRINT
"SORRY-INCORRECT"
3165 IF OPER$="><" THEN PRINT "><  INDICATES A TWO TAILED
TEST" ELSE PRINT "< OR > INDICATE A ONE TAILED TEST"
3168 FOR I= 1 TO 1200:NEXT I:CLS
3170 PRINT "STEP 3. DETERMINE THE LEVEL OF SIGNIFICANCE
```

```
(ALPHA) AS STATED IN THE PROBLEM"
3175 PRINT:INPUT "INPUT LEVEL OF SIG. (always a decimal
value)";LSIG:S2=LSIG/R%
3180 IF LSIG<0 OR LSIG>1 THEN GOTO 3175
3190 CLS:PRINT"STEP 4. DETERMINE FROM THE TABLE THE PROPER
T-ALPHA. REMEMBER TO USE THE RIGHT      DEGREES OF FREEDOM
(N-2)":PRINT
3195 PRINT "FIND THE T-ALPHA CORRESPONDING
TO";LSIG/R%;"WITH";SNUM-2;"DEGREES OF FREEDOM"
3200 INPUT "INPUT T-ALPHA VALUE";TALPHA
3210 GOSUB 15000
3215 FOR I=1 TO 500:NEXT I
3220 CLS:PRINT"STEP 5. THE NEXT STEP IS TO COMPUTE THE TEST
STATISTIC. (Formula-15)"
3221 PRINT "PUSH ENTER AND THE COMPUTER  WILL DO IT FOR
YOU."
3225 INPUT Q
3230 CLS:RETURN
3480 '****************************************
3485 'DETERMINE WHETHER TO ACCEPT OR REJECT THE NULL
HYPOTHESIS
3490 '****************************************
3500 IF (OPER$="<") AND (TSTAT<-TALPHA) THEN
FLAG$="REJECT":RETURN
3510 IF (OPER$=">") AND (TSTAT>TALPHA) THEN
FLAG$="REJECT":RETURN
3520 IF (OPER$="><") AND ((TSTAT<-TALPHA) OR (TSTAT>TALPHA))
THEN FLAG$="REJECT":RETURN
3530 FLAG$="ACCEPT"
3540 RETURN
3580 '****************************************
3585 'OUTPUT ROUTINE
3590 '****************************************
3600 CLS
3605 PRINT"6. BASED ON THE RELATIONSHIP BETWEEN THE COMPUTED
TEST STATISTIC AND THE T-ALPHA VALUE OBTAINED FROM THE CHART
DETERMINE IF YOU CAN REJECT THE NULL HYPOTHESIS."
3610 PRINT"REMEMBER, T-ALPHA=";TALPHA;"TEST STAT=";TSTAT
3620 INPUT "R-TO REJECT OR A-FOR ACCEPT";R$:CLS
3625 IF R$<>"R" AND R$<>"A" THEN GOTO 3610
3630 IF R$="R" AND FLAG$="REJECT" THEN PRINT
"CORRECT":PRINT"WE SHOULD REJECT THE NULL HYP.":RETURN
3635 IF R$="A" AND FLAG$="ACCEPT" THEN PRINT
"CORRECT":PRINT"WE CANNOT REJECT THE NULL HYP.":RETURN
3640 PRINT"WRONG"
3645 PRINT"THESE VALUES SUGGEST WE SHOULD ";FLAG$
3650 PRINT"BETTER LUCK NEXT TIME":RETURN
3970 '========================================
3975 '========================================
3980 '****************************************
```

```
3985 'MAIN PROGRAM AND BASIC DATA ENTRY FOR CONFIDENCE
INTERVALS ABOUT ALPHA OR BETA
3990 '*****************************************
4000 CLS:PRINT"*CONFIDENCE INTERVAL (for alpha or
beta)****************************************"
4010 PRINT"CI FOR alpha OR beta?"
4030 INPUT "INPUT A-FOR alpha:B FOR beta";SWITCH$
4040 IF SWITCH$<>"A" AND SWITCH$<>"B" THEN PRINT"ENTER
CAPITAL A OR B!":GOTO 4010
4050 CLS:PRINT"STEP 1. DETERMINE THE DEGREE OF
CONFIDENCE REQUIRED(usually stated as a % in the
problem)":PRINT
4060 INPUT "INPUT THIS VALUE (in decimal form)";DCNF
4070 IF DCNF>1 OR DCNF<0 THEN GOTO 4060
4080 CLS:PRINT"STEP 2. CONVERT THE DEGREE OF CONF.
REQUIRED TO A LEVEL OF SIGNIFICANCE      (ALPHA)
(ALPHA=1-D.ofC.)":PRINT
4085 INPUT"INPUT THE LEVEL OF SIG.(always a decimal
value!";LSIG
4095 S2=LSIG/2
4100 IF LSIG<>1-DCNF OR LSIG >1 OR LSIG<0 THEN
PRINT"WRONG!:YOU CHECK YOUR MATH AND TRY AGAIN":GOTO 4085
4110 CLS:PRINT"STEP 3. USING TABLE II, FIND THE PROPER
T-ALPHA VALUE. (remember the degrees of freedom=(n-2):in
this case DFRD=";SNUM-2;")(also since all CIs are two
tailed, divide your level of sig. by 2:in this case
LSIG=";S2;")"
4120 INPUT "ENTER THE TABLE VALUE(TALPHA)";TALPHA
4130 GOSUB 15000
4140 FOR I=1 TO 500:NEXT I
4150 CLS:PRINT"COMPUTE THE CI INTERVAL.(use
formula-16)":PRINT
4160 INPUT"PUSH ENTER AND THE COMPUTER WILL DO IT  FOR
YOU!!";Q
4170 GOSUB 8000
4175 '**********OUTPUT ROUTINE**********
4180 CLS:PRINT
4190 IF SWITCH$="A"THEN PRINT ALOWER;"< alpha <";AUPPER ELSE
PRINT BLOWER;"< BETA <";BUPPER
4200 PRINT:INPUT"SELECT: 1-MENU(same data); 2-MORE CIs;
3-NEW DATA ENTRY; 4-EXIT";Q
4210 ON Q GOTO 1090,4000,900,1180
4220 END
5070 '========================================
5075 '========================================
5080 '*****************************************
5085 'MAIN PROGRAM AND DATA ENTRY FOR CONFIDENCE INTERVAL
ABOUT THE MEAN OF Yo
5090 '*****************************************
5100 CLS:PRINT"**CI FOR THE MEAN OF Yo WHEN GIVEN Xo**"
```

```
5110 PRINT"****************************************"
5120 PRINT"VALUES NEEDED:","STD.ERROR"
5130 PRINT"T-ALPHA VALUE","A and B"
5140 PRINT"X-SUM","X-SUM^2"
5150 PRINT"X-MEAN","Xo"
5160 INPUT "PUSH ENTER TO CONTINUE";Q
5170 CLS:PRINT"THE ONLY NEW VALUES TO DETERMINE ARE THE
VALUE OF Xo AND THE PROPER T-ALPHA     VALUE":PRINT
5180 PRINT"STEP 1. EXTRACT THE VALUE FOR Xo FROM   THE
PROBLEM.":PRINT
5190 INPUT "INPUT Xo";XNAUGHT
5200 CLS:PRINT"STEP 2. DETERMINE THE LEVEL OF
SIGNIFICANCE FROM THE GIVEN DEGREE OF  CONFIDENCE REQUIRED.
(remember the      L.O.S.= 1 - the deg. of conf.)":PRINT
5210 INPUT "ENTER THE LEV. OF SIG.(always a decimal
value!)";LSIG
5220 IF LSIG>1 OR LSIG<0 THEN GOTO 5210
5225 S2=LSIG/2
5230 CLS:PRINT"STEP 3. USING TABLE II, FIND THE PROPER
T-ALPHA VALUE. (remember the degrees of freedom=(N-2):in
this case DFRD=";SNUM-2;")(also, since all CIs are two
tailed tests, divide your level of sig. by two:in this case
L.O.S.=";S2;")":PRINT
5240 INPUT "ENTER THE TABLE VALUE T-ALPHA";TALPHA
5250 GOSUB 15000
5260 FOR I=1 TO 500:NEXT I
5270 CLS:PRINT"STEP 4. COMPUTE THE CONFIDENCE INTERVAL USING
FORMULA-17.":PRINT
5280 INPUT "PUSH ENTER AND THE COMPUTER WILL ASSIST!";Q
5290 GOSUB 9750
5300 CLS:PRINT:PRINT ULOWER;"< Yo <";UUPPER:PRINT
5310 INPUT "SELECT: 1-MENU (same data); 2-NEW DATA  ENTRY;
3-EXIT";Q
5320 ON Q GOTO 1090,900,1180
5330 END
5970 '========================================
5975 '========================================
5980 '****************************************
5985 'MAIN PROGRAM AND DATA ENTRY FOR THE COMPUTATION OF THE
LIMITS OF PREDICTION
5990 '****************************************
6000 CLS:PRINT"*********LIMITS OF PREDICTION************"
6010 PRINT"THE COMPUTATIONAL STEPS ARE IDENTICAL TO THE
DEVELOPMENT OF A CI ABOUT THE MEAN OF Y. (in this case
however, formula-18 is used)":PRINT
6020 INPUT "PUSH ENTER TO CONTINUE";Q
6030 CLS:PRINT"STEP 1. DETERMINE Xo FROM THE PROBLEM.":PRINT
6040 INPUT"INPUT Xo";XNAUGHT
6050 CLS:PRINT"STEP 2. DETERMINE THE LEVEL OF
SIGNIFICANCE NEEDED. (remember          L.O.S.=(1-degree of
confidence)":PRINT
6060 INPUT "INPUT THE LEV. OF SIG.(always a decimal
value!)";LSIG
6065 IF LSIG>1 OR LSIG<0 THEN GOTO 6060
6070 S2=LSIG/2
6080 CLS:PRINT"STEP 3. FIND THE PROPER T-ALPHA VALUE   FROM
TABLE II. (degrees of freedom=(n-2) or in this
case";SNUM-2;"(remember to  account for both tails you must
divide  the LEV. OF SIG. BY TWO, in this case
L.O.S./2=";S2;")"
6090 INPUT "INPUT T-ALPHA";TALPHA
6100 GOSUB 15000
6110 FOR I=1 TO 500:NEXT I
6120 CLS:PRINT"STEP 4. THE FINAL STEP IS TO COMPUTE    THE
INTERVAL USING THE PROPER FORMULA.  (Formula-18)"
6130 GOSUB 9850
6135 '**************OUTPUT**************
6140 PRINT"THE PREDICTION LIMITS ARE:"
6150 PRINT"LOWER=";LLOWER;"UPPER=";LUPPER:PRINT
6160 INPUT"SELECT: 1-MENU(same data set); 2-INPUT NEW DATA;
3-EXIT";Q
6170 ON Q GOTO 1090,900,1180
6970 '========================================
6975 '========================================
6980 '****************************************
6985 'MAIN PROGRAM AND DATA ENTRY TO FIND AN ESTIMATE OF Y
WHEN GIVEN X
6990 '****************************************
7000 CLS:PRINT"*******ESTIMATES OF Y WHEN GIVEN X******"
7010 PRINT"SOLUTION:SOLVE THE LEAST-SQUARES LINE   EQUATION
WITH THE GIVEN INPUT FOR X."
7020 PRINT"Y-hat=";A;"+";B;"X"
7030 INPUT "INPUT X";Q5
7040 PRINT"Y=";A+B*Q5
7050 FOR I=1 TO 1600:NEXT I:GOTO 1090
7970 '========================================
7975 '========================================
7980 '****************************************
7985 'COMPUTATION THE C.I. ABOUT A AND B
7990 '****************************************
8000 ACNF=TALPHA*SE*SQR((1/N)+(N*XMEAN2)/(N*X2SUM-XSUM^2))
8020 ALOWER=A-ACNF:AUPPER=A+ACNF
8030 BCNF=(TALPHA*SE)/SQR((N*X2SUM-XSUM^2)/N)
8040 BLOWER=B-BCNF:BUPPER=B+BCNF
8050 RETURN
8980 '****************************************
8985 'COMPUTATION AND DISPLAY OF THE TEST STATISTIC FOR HYP.
TEST OF ALPHA
8990 '****************************************
9000 ATSTAT=(A-GALPHA)/(SE
*SQR((1/N)+(N*XMEAN2)/(N*X2SUM-XSUM^2))):TSTAT=ATSTAT
9010 CLS:PRINT"GIVEN THE FOLLOWING INPUTS:"
9020 PRINT"NULL HYP =";NHYP;"  SAM SIZE =";SNUM
9030 PRINT"STD.ERR. =";SE;"  X-SUM =";XSUM
9040 PRINT"X^2-SUM=";X2SUM;"XMEAN=";MEAN
9050 PRINT"THE TEST STATISTIC IS ="TSTAT
9060 FOR I=1 TO 1500:NEXT I
9070 RETURN
9480 '****************************************
9485 'COMPUTATION AND DISPLAY OF THE TEST STATISTIC FOR HYP.
TEST OF BETA
9490 '****************************************
9500
BTSTAT=((B-GBETA)/SE)*SQR((N*X2SUM-XSUM^2)/N):TSTAT=BTSTAT
9510 CLS:PRINT"GIVEN THE FOLLOWING INPUTS:"
9520 PRINT"NULL HYP =";NHYP;"  SAM SIZE =";SNUM
9530 PRINT"STD.ERR. =";SE;"  X-SUM =";XSUM
9540 PRINT"X^2-SUM=";X2SUM;"XMEAN=";MEAN
9550 PRINT"THE TEST STATISTIC IS ="TSTAT
9560 FOR I=1 TO 1500:NEXT I
9570 RETURN
9730 '****************************************
9735 'COMPUTATION OF THE C.I. ABOUT THE MEAN OF Yo
9740 '****************************************
9750
UCNF
TALP
A*SE*SQR((1/N)+((N*((XNAUGHT-MEAN)^2)))/(N*X2SUM-XSUM^2))
9760 ULOWER=(A+B*XNAUGHT)-UCNF:UUPPER=(A+B*XNAUGHT)+UCNF
9770 RETURN
9830 '****************************************
9835 'COMPUTATION OF THE PREDICTION LIMITS
9840 '****************************************
9850
LCNF
TALP
A*SE*SQR(1+(1/N)+(N*((XNAUGHT-MEAN)^2))/(N*X2SUM-XSUM^2))
9860 LLOWER=(A+B*XNAUGHT)-LCNF:LUPPER=(A+B*XNAUGHT)+LCNF
9870 RETURN
14980 '****************************************
14985 'COMMON TABLE LOOK-UP ROUTINE
14990 '****************************************
15000 IF S2=.1 THEN PTR%=1
15005 IF S2=.05 THEN PTR%=6
15010 IF S2=.025 THEN PTR%=11
15015 IF S2=.010 THEN PTR%=16
15020 IF S2=.005 THEN PTR%=21
15025 IF PTR%=0 THEN PRINT "ODD LEVEL OF SIGNIFICANCE: NO
ERROR     CHECKING AVAILABLE":RETURN
15030 IF SNUM>31 THEN TNUM=31 ELSE TNUM=SNUM
```

```
15035 TVL$= MID$(TABLE$(TNUM-3),PTR%,5)
15040 TL=VAL(TVL$)
15045 IF TALPHA<>TL THEN PRINT "WRONG VALUE TRY AGAIN!!"
ELSE PRINT "CORRECT":RETURN
15050 INPUT "INPUT T-ALPHA VALUE (LAST CHANCE)";TALPHA
15055 CLS
15060 IF TL<>TALPHA THEN PRINT "WRONG AGAIN!! THEN CORRECT
VALUE IS";TL:TALPHA=TL ELSE PRINT "RIGHT ON"
15065 RETURN
19980 '****************************************
19985 'TABLE DATA
19990 '****************************************
20000 DEFSNG A-Z
20020 ON ERROR GOTO 22000
20055 DIM TABLE$(29)
20060 FOR L=0 TO 29
20065   READ TABLE$(L)
20070 NEXT L
20071 DATA "3.0786.31412.7031.8263.65"
20072 DATA "1.8862.9204.3036.9659.925"
20073 DATA "1.6382.3533.1824.5415.841"
20074 DATA "1.5332.1322.7763.7474.604"
20075 DATA "1.4762.0152.5713.3654.032"
20076 DATA "1.4401.9432.4473.1433.707"
20077 DATA "1.4151.8952.3652.9983.499"
20078 DATA "1.3971.8602.3062.8963.355"
20079 DATA "1.3831.8332.2622.8213.250"
20080 DATA "1.3721.8122.2282.7643.169"
20081 DATA "1.3631.7962.2012.7183.106"
20082 DATA "1.3561.7822.1792.6813.055"
20083 DATA "1.3501.7712.1602.6503.012"
20084 DATA "1.3451.7612.1452.6242.977"
20085 DATA "1.3411.7532.1312.6022.947"
20086 DATA "1.3371.7462.1202.5832.921"
20087 DATA "1.3331.7402.1102.5672.898"
20088 DATA "1.3301.7342.1012.5522.878"
20089 DATA "1.3281.7292.0932.5392.861"
20090 DATA "1.3251.7252.0862.5282.845"
20091 DATA "1.3231.7212.0802.5182.831"
20092 DATA "1.3211.7172.0742.5082.819"
20093 DATA "1.3191.7142.0692.5002.807"
20094 DATA "1.3181.7112.0642.4922.797"
20095 DATA "1.3161.7082.0602.4852.787"
20096 DATA "1.3151.7062.0562.4792.779"
20097 DATA "1.3141.7032.0522.4732.771"
20098 DATA "1.3131.7012.0482.4672.763"
20099 DATA "1.3111.6992.0452.4622.756"
20100 DATA "1.2821.6451.9602.3262.576"
20105 RETURN
21980 '****************************************
```

```
21985 'ERROR TRAP
21990 '****************************************
22000 PRINT "ERROR";ERR;"IN LINE";ERL
22010 PRINT"CONSULT OPERATORS MANUAL":END
```

```
0 'TITLE: MODULE 8
1 'DATE: 30 APR 85
2 'PURPOSE: THIS PROGRAM IS DESIGNED TO ASSIST THE STUDENT
IN SOLVING MULTIPLE REGRESSION AND CORRELATION PROBLEMS.
3 'SPECIFIC APPLICATION SUBROUTINES INCLUDE: SIGNIFICANCE
TESTING FOR THE COEFFICIENT OF CORRELATION; CALCULATION OF
THE COEFFICIENT OF DETERMINATION; AND ESTIMATION OF THE
VALUE OF Y WHEN GIVEN THE CORRESPONDING X VALUE(S).
4 '
5 'SIGNIFICANT SUBROUTINES WHICH ARE USED JOINTLY BY ALL THE
ABOVE APPLICATIONS INCLUDE:
6 'A DATA INPUT ROUTINE WHICH FORMATS AND DISPLAYS THE INPUT
DATA IN TABULAR FORM.
7 'A DATA CORRECTION ROUTINE WHICH ALLOWS FOR MODIFICATION
OF INPUT DATA.
8 'TWO ROUTINES WHICH HANDLE THE BASIC CALCULATION OF THE
LEAST-SQUARES LINE (AND OTHER PERTINENT VALUES); ONE FOR THE
CASE OF TWO VARIABLES AND ANOTHER FOR INSTANCES OF THREE
VARIABLES.
9 'A DATA TABLE ROUTINE WHICH PROVIDES R VALUES AND A
ROUTINE WHICH CHECKS A STUDENT'S INPUT FOR 'R' TO THE DATA
TABLE.
10 '
11 'INPUT VARIALBES
12 'A2=ARRAY OF RAW DATA VALUES
14 'LSIG=LEVEL OF SIGNIFICANCE
16 'NVAR=NUMBER OF VARIABLES IN THE PROBLEM
18 'OPAIRS=NUMBER OF ORDERED PAIRS IN THE SAMPLE
20 'R$=STUDENT'S ACCEPT/REJECT CRITERIA
22 'RC$=REQUEST TO REVIEW INPUT DATA
24 'R2C2$=REQUEST TO MODIFY DATA
26 'R3C4$=REQUEST TO MODIFY MORE DATA
28 'TR=STUDENT'S TABLE LOOK-UP VALUE
30 'UPDATE=NEW DATA VALUE
32 '
34 'OUTPUT VARIABLES
36 'A=COEFFICIENT IN LEAST-SQUARES LINE
38 'A1/B = ARRAYS WHICH CONTAIN VALUES OF THE NORMAL
EQUATIONS
40 'B=COEFFICIENT IN LEAST-SQUARES LINE
41 'C=ARRAY OF COEFFICIENTS OF THE LEAST-SQUARES LINE
42 'FLAG$=ACCEPT/REJECT DESIGNATOR
44 'R=COEFFICIENT OF CORRELATION
46 'SE= STANDARD ERROR
48 '
50 'INTERNAL VARIABLES
52 'I:J:K:L = LOOP INDEX (J is also used as a column
indicator in the Data Update routine)
54 'MEAN=AVG. OF INPUT X VALUES
56 'N=NUMBER OF ORDERED PAIRS IN THE INPUT DATA SET
```

```
58 'PTR%=POINTER
60 'Q=FLOW CONTROL (also used is an intermediate value in
the Least-Squares calulation)
62 'Q1:Q2:Q3=SCREEN FORMAT VARIABLES
64 'TABLE=COMMON TABLE R VALUES
68 'TL=TABLE R VALUE
70 'TVL$=INTERMEDIATE TABLE VALUE
72 'XSUM=SUM OF ALL INPUT X VALUES
74 'XYSUM=SUM OF ALL X*Y VALUES
76 'X2SUM=SUM OF ALL X-SQUARED VALUES
78 'YHAT=POINT VALUE OF THE LEAST-SQUARES LINE
80 'YMEAN=AVG. OF ALL THE INPUT Y VALUES
82 'YSUM=SUM OF ALL THE INPUT Y VALUES
84 'Y2SUM=SUM OF ALL THE Y-SQUARED VALUES
86 '
470 '****************************************
480 'PROGRAM INTRODUCTION AND DATA INPUT DRIVER
490 '****************************************
500 CLS:PRINT"****MULTIPLE REGRESSION + CORRELATION***";
510 PRINT"NOTE:THIS PROGRAM CAN BE USED FOR SIMPLE
REGRESSION (two variables) AS WELL AS  MULTIPLE REGRESSION
(with three          variables).":PRINT
520 INPUT "PUSH ENTER TO CONTINUE";Q
530 CLS:PRINT"THE LEAST-SQUARES LINE FOR DATA SETS     WITH
MORE THAN TWO VARIABLES IS OBTAINED BY SOLVING A SYSTEM OF
NORMAL EQUATIONS SIMULTANEOUSLY. THESE NORMAL EQUATIONS ARE
EASILY OBTAINED FROM THE DATA (using formulas-19, 20, and
21)."
540 PRINT"THE FIRST STEP IS TO INPUT THE DATA."
550 INPUT "PUSH ENTER TO CONTINUE!";Q
560 CLEAR:GOSUB 20000:REM ***INITIALIZE TABLE VALUES***
570 GOSUB 1000
580 INPUT"WOULD YOU LIKE TO REVIEW OR CHANGE ANY OF YOUR
INPUT DATA?(YorN)";RC$
590 IF RC$="Y" OR RC$="y" THEN GOSUB 1500
600 IF NVAR=3 THEN GOSUB 2050 ELSE GOSUB 3100
610 FOR I=1 TO 1600:NEXT I
613 '****************************************
615 'TOTAL PROGRAM MENU SELECTION
617 '****************************************
620 CLS:PRINT TAB(18);"MENU"
630 PRINT"1-SIGNIFICANCE FOR r"
640 PRINT"2-COEFFICIENT OF DETERMINATION"
650 PRINT"3-ESTIMATES OF Y WHEN GIVEN X (X1,X2)
655 PRINT"4-EXIT":PRINT
660 INPUT "SELECTION";Q
670 ON Q GOTO 4000,5000,6000,690
680 END
690 MENU
970 '****************************************
```

```
980 'DATA INPUT AND DISPLAY ROUTINE
990 '*****************************************
1000 CLS:INPUT "INPUT THE NUMBER OF VARIABLES (2 or
3).";NVAR
1010 IF NVAR<>2 AND NVAR<>3 THEN PRINT"THE PROGRAM ONLY
WORKS FOR 2 OR 3 VARIABLES!!":GOTO 1000
1020 INPUT "INPUT THE NUMBER OF OBSERVATIONS.";OPAIRS:CLS
1025 IF OPAIRS<3 THEN PRINT"YOU NEED AT LEAST 3
OBSERVATIONS":GOTO 1020
1026 IF INT(ABS(OPAIRS))<>OPAIRS THEN PRINT"MUST BE A
POSITIVE INTEGER!":GOTO 1020
1030 DIM A2(OPAIRS,3)
1035 Q1=0:Q2=17:Q3=29
1040 FOR I=1 TO OPAIRS
1050  PRINT @Q1,I;". X1=";
1060  INPUT A2(I,1)
1070  IF NVAR=3 THEN PRINT @Q2,"X2=";:INPUT A2(I,2)
1080  PRINT @Q3,"Y=";
1090  INPUT A2(I,3)
1100  Q1=Q1+40:Q2=Q2+40:Q3=Q3+40
1110  IF Q1>240 THEN CLS:Q1=0:Q2=16:Q3=26
1120 NEXT I
1130 PRINT"DATA INPUT IS COMPLETE!!"
1140 RETURN
1470 '*****************************************
1480 'DATA REVIEW AND UPDATE ROUTINE
1490 '*****************************************
1500 CLS:PRINT"YOUR DATA WILL SCROLL DOWN THE SCREEN   USE
THE PAUSE KEY IF NEEDED!"
1505 INPUT "PUSH ENTER WHEN READY";Q
1510 FOR I=1 TO OPAIRS
1515 PRINT I;":";
1520  IF NVAR=3 THEN PRINT USING
"#######.##";A2(I,1);A2(I,2);A2(I,3) ELSE PRINT USING
"###########.###";A2(I,1);A2(I,3)
1525 FOR J=1TO300:NEXT J
1530 NEXT I
1540 INPUT "WOULD YOU LIKE TO CHANGE ANY DATA(YorN)";R2C2$
1550 IF R2C2$="N" OR R2C2$="n" THEN RETURN
1555 IF R2C2$<>"Y" AND R2C2$<>"y" THEN GOTO 1540
1560 INPUT "IN WHICH COLUMN DOES THE UPDATE GO?(1,2 or 3)";J
1570 INPUT "WHICH OBSERVATION?(ie row number of the element
to be changed:1,2,3,etc)";I
1580 IF NVAR=2 AND J=2 THEN J=3
1585 PRINT"CURRENT VALUE =";A2(I,J)
1590 INPUT "INPUT THE NEW VALUE";UPDATE
1600 A2(I,J)=UPDATE
1610 INPUT "ANY MORE CHANGES(Y or N)";R3C4$
1620 IF R3C4$="Y" OR R3C4$="y" THEN GOTO 1560
1630 RETURN
```

```
2030 '****************************************
2040 'CALCULATION OF THE NORMAL AND LEAST-SQUARES EQUATIONS
(for three variables)
2045 '****************************************
2050 DIM A1(3,3)
2100 FOR I=1 TO OPAIRS
2110  B(1)=B(1)+A2(I,3)
2120  B(2)=B(2)+A2(I,1)*A2(I,3)
2130  B(3)=B(3)+A2(I,2)*A2(I,3)
2140  A1(2,1)=A1(2,1)+A2(I,1)
2150  A1(3,1)=A1(3,1)+A2(I,2)
2160  A1(2,2)=A1(2,2)+A2(I,1)^2
2170  A1(3,3)=A1(3,3)+A2(I,2)^2
2180  A1(2,3)=A1(2,3)+A2(I,1)*A2(I,2)
2190 NEXT I
2200 A1(1,2)=A1(2,1):A1(1,3)=A1(3,1):A1(3,2)=A1(2,3)
2201 CLS:PRINT"THE NORMAL EQUATIONS ARE:"
2202 PRINTB(1);"=";OPAIRS;"b0 +";A1(2,1);"b1 +";A1(3,1);"b2"
2203 PRINTB(2);"=";A1(2,1);"b0 +";A1(2,2);"b1 +";A1(2,3)"b2"
2204 PRINTB(3);"=";A1(3,1);"b0 +";A1(3,2);"b1
+";A1(3,3);"b2"
2205 YMEAN=B(1)/OPAIRS:A1(1,1)=OPAIRS
2207 '******CALCULATION OF THE LEAST SQUARES
COEFFICIENTS******
2210 FOR I=1 TO 3
2220  Q=A1(I,I)
2230  IF Q<>0 THEN 2250
2240  PRINT"SOLUTION NOT POSSIBLE!!":END
2250  A1(I,I)=1
2260  FOR J=1 TO 3
2270   A1(I,J)=A1(I,J)/Q
2280  NEXT J
2290  FOR K=1 TO 3
2300   IF K=I THEN 2360
2310   Q=A1(K,I)
2320   A1(K,I)=0
2330   FOR J=1 TO 3
2340    A1(K,J)=A1(K,J)-Q*A1(I,J)
2350   NEXT J
2360  NEXT K
2370 NEXT I
2400 FOR I= 1 TO 3
2410  C(I)=0
2420  FOR K=1 TO 3
2430   C(I)=C(I)+A1(I,K)*B(K)
2440  NEXT K
2450 NEXT I
2500 FOR I=1 TO OPAIRS
2510  YHAT=C(1)+C(2)*A2(I,1)+C(3)*A2(I,2)
2520  TPSUM=(A2(I,3)-YHAT)^2+TPSUM
```

```
2530  BTTM=(A2(I,3)-YMEAN)^2+BTTM
2540 NEXT I
2550 IF OPAIRS<4 THEN SE=-999 ELSE SE=SQR(TPSUM/(OPAIRS-3))
2560 R=SQR(1-TPSUM/BTTM)
2570 PRINT"THE SOLUTION TO THESE EQUATIONS IS THE
LEAST-SQUARES LINE:"
2580 PRINT"Yhat=";C(1);"+";C(2);"X1 +";C(3);"X2"
2590 RETURN
3080 '****************************************
3090 'CALCULATION OF THE LEAST-SQUARES LINE (for two
variables)
3095 '****************************************
3100 FOR I=1 TO OPAIRS
3110  XSUM=XSUM+A2(I,1)
3120  X2SUM=X2SUM+A2(I,1)^2
3130  YSUM=YSUM+A2(I,3)
3140  Y2SUM=Y2SUM+A2(I,3)^2
3150  XYSUM=XYSUM+A2(I,1)*A2(I,3)
3160 NEXT I
3170 N=OPAIRS:MEAN=XSUM/N:XMEAN2=MEAN^2
3180 B=(N*XYSUM-XSUM*YSUM)/(N*X2SUM-XSUM^2)
3190 A=(YSUM-B*XSUM)/N:SE=SQR((Y2SUM-A*YSUM-B*XYSUM)/(N-2))
3200
R=(N
XYSU
-XSUM*YSUM)/(SQR(N*X2SUM-(XSUM^2))*SQR(N*Y2SUM-(YSUM^2)))
3210 CLS:PRINT"THE LEAST-SQUARES LINE="
3220 PRINT "Y-hat=";A;"+";B;"X"
3230 PRINT"STANDARD ERROR=";SE
3240 RETURN
3970 '========================================
3975 '========================================
3980 '****************************************
3985 'INTRODUCTION TO SIGNIFICANCE TEST FOR THE COEFFICIENT
OF CORRELATION
3990 '****************************************
4000 CLS:PRINT"***********SIGNIFICANCE TEST************";
4010 PRINT"**FOR THE COEFFICIENT OF CORRELATION(r)*";
4020 PRINT"OBJECT: TO DETERMINE IF 'r' MAY BE
ATTRIBUTED TO CHANCE AT A CERTAIN LEVEL OF
SIGNIFICANCE!":PRINT
4025 INPUT "PUSH ENTER TO CONTINUE!";Q
4030 CLS:PRINT"TO BEGIN, ASSUME A NULL HYPOTHESIS OF   'NO
CORRELATION'(meaning the r value is totally attributed to
chance and there  is no correlation between y and the
corresponding x value(s).)"
4040 PRINT"NOW, LET'S TEST THE NULL HYP. TO SEE IF IT CAN BE
REJECTED."
4045 INPUT "PUSH ENTER TO CONTINUE!";Q
4050 GOSUB 4500
```

```
4060 PRINT:INPUT "SELECT: 1-MENU (same data); 2-MORE
SIGNIFICANCE TESTING; 3-ENTER NEW DATA  SET; 4-EXIT";Q
4070 ON Q GOTO 620,4030,560,690
4080 GOTO 620
4480 '****************************************
4485 'MAIN PROGRAM BODY
4490 '****************************************
4500 CLS:PRINT"STEP 1. DETERMINE THE LEVEL OF
SIGNIFICANCE (ALPHA).":PRINT
4510 INPUT "ENTER THE L.O.S.(ALPHA)(always a decimal
value)";LSIG
4520 IF LSIG>1 OR LSIG<=0 THEN PRINT"ALPHA MUST BE IN
DECIMAL FORM!!":GOTO 4510
4530 CLS:PRINT"STEP 2. SINCE THIS PROCEDURE CALLS FOR  A TWO
TAILED TEST, DIVIDE THE LEVEL OF  SIG. (ALPHA) BY
TWO.":PRINT
4540 PRINT"(REMEMBER: ALPHA=";LSIG;")"
4545 INPUT "INPUT ALPHA/TWO";S2
4550 IF S2<>LSIG/2 THEN PRINT"WRONG!!TRY AGAIN!":GOTO 4540
4560 CLS:PRINT"STEP 3. DETERMINE FROM TABLE VI THE
PROPER r-(ALPHA/2) VALUE.":PRINT
4570 PRINT"(REMEMBER: ALPHA/2=";S2;"AND THE NUMBER OF
OBSERVATIONS=";OPAIRS;")"
4580 INPUT "ENTER THE TABLE VALUE!";TR
4590 GOSUB 15000:FOR I=1 TO 500:NEXT I:REM *CHECK INPUT
AGAINST TABLE DATA
4600 CLS:PRINT"STEP 4. COMPUTE THE COEFFICIENT OF
CORRELATION (r). (use formula-22 or formula 23 as
applicable)":PRINT
4610 INPUT "PUSH ENTER AND THE COMPUTER WILL DO IT  FOR
YOU!!";Q
4620 PRINT"r=";R:FOR I=1 TO 500:NEXT I
4625 '***DETERMINE WHETHER TO ACCEPT/REJECT***
4630 CLS:PRINT"STEP 5. BASED ON THE RELATIONSHIP
BETWEEN THE COMPUTED VALUE FOR r AND    THE r-ALPHA/2 VALUE
OBTAINED FROM THE   TABLE, DETERMINE IF YOU CAN REJECT THE
NULL HYPOTHESIS."
4640 PRINT"REMEMBER:r-ALPHA/2=";TR
4650 PRINT TAB(9);"COMPUTED VALUE OF r =";R
4660 IF TR<ABS(R) THEN FLAG$="REJECT" ELSE FLAG$="ACCEPT"
4670 INPUT "R-TO REJECT OR A-TO ACCEPT";R$
4675 '**********OUTPUT RESULTS*********
4680 CLS:IF R$<>"R" AND R$<>"A" THEN PRINT"INPUT MUST BE
UPPER CASE:R OR A":GOTO 4670
4690 IF R$="R" AND FLAG$="REJECT" THEN PRINT "CORRECT! WE
SHOULD REJECT THE NULL      HYPOTHESIS:(ie: There is a
relationship between Y and X (or Y and X1,X2)":RETURN
4700 IF R$="A" AND FLAG$="ACCEPT" THEN PRINT"CORRECT! WE
CANNOT REJECT THE NULL      HYPOTHESIS:(ie: The calculated
value of r may be due to chance.)":RETURN
```

```
4710 PRINT"WRONG!!"
4720 PRINT"THESE VALUES SUGGEST WE SHOULD ";FLAG$;" THE NULL
HYP."
4730 PRINT"BETTER LUCK NEXT TIME!":RETURN
4970 '========================================
4975 '========================================
4980 '****************************************
4985 'CALCULATION OF THE COEFFICIENT OF DETERMINATION
4990 '****************************************
5000 CLS:PRINT"******COEFFICIENT OF DETERMINATION******";
5010 PRINT"OBJECT: DETERMINE HOW MUCH OF THE TOTAL VARIATION
AMONG THE Y VALUES IN A DATA  SET CAN BE ATTRIBUTED TO A
RELATIONSHIP WITH THE OTHER VARIABLES IN THE DATA    SET AND
HOW MUCH IS DUE TO CHANCE.":PRINT
5020 INPUT "PUSH ENTER TO CONTINUE!";Q
5030 CLS:PRINT"STEP 1. CALCULATE THE COEFFICIENT OF
CORRELATION (r). (use  formula-22 or formula-23 as
applicable)":PRINT
5040 INPUT "PUSH ENTER AND THE COMPUTER WILL DO IT  WITH
YOU!!";Q
5050 PRINT"r=";R:FOR I=1 TO 600:NEXT I
5060 CLS:PRINT"STEP 2. SQUARE THE VALUE OF r TO OBTAIN THE
COEFFICIENT OF DETERMINATION. (THE  CO. of DET. TIMES 100
GIVES THE PERCENT OF THE TOTAL VARIATION OF Y WHICH IS
EXPLAINED BY THE RELATIONSHIP WITH X    (or X1 and X2)"
5070 PRINT"r SQUARED * 100=";(R^2)*100;"%"
5080 INPUT "PUSH ENTER TO CONTINUE!";Q
5090 CLS:PRINT"THAT'S ALL THERE IS TO IT!!":PRINT
5100 INPUT "SELECT: 1-MENU(same data set);2-ENTER   NEW DATA
SET:3-EXIT";Q
5110 ON Q GOTO 620,560,690
5120 GOTO 620
5970 '========================================
5975 '========================================
5980 '****************************************
5985 'CALCULATE Y WHEN GIVEN X OR (X1,X2)
5990 '****************************************
6000 CLS:PRINT"**ESTIMATES OF Y WHEN GIVEN X (X1,X2)**"
6010 PRINT"*****************************************";
6020 PRINT"SOLUTION:  SOLVE THE LEAST-SQUARES LINE EQUATION
WITH THE GIVEN INPUT FOR X (or X1,X2).":PRINT
6030 IF NVAR>2 THEN GOTO 6100
6040 PRINT"Y-hat=";A;"+";B;"X"
6050 INPUT "INPUT X";Q
6060 PRINT"Y=";A+B*Q
6070 FOR I=1 TO 1600:NEXT I:GOTO 620
6100 PRINT"Y-hat=";C(1);"+";C(2);"X1 +";C(3);"X2"
6110 INPUT "INPUT X1";X1:INPUT "INPUT X2";X2
6120 PRINT"Y=";C(1)+C(2)*X1+C(3)*X2
6130 FOR I=1 TO 1600:NEXT I:GOTO 620
```

```
14970 '========================================
14975 '========================================
14980 '****************************************
14985 'DATA TABLE INPUT
14990 '****************************************
15000 PTR%=0
15010 IF S2=.025 THEN PTR%=1
15020 IF S2=.010 THEN PTR%=4
15030 IF S2=.005 THEN PTR%=7
15040 IF PTR%=0 THEN PRINT"ODD LEVEL OF SIGNIFICANCE: NO
ERROR     CHECKING AVAILABLE!":RETURN
15050 IF OPAIRS>22 THEN PRINT"NO ERROR CHECKING AVAILABLE
FOR A SAMPLE SIZE GREATER THAN 22!":RETURN
15060 TVL$=MID$(TABLE$(OPAIRS-3),PTR%,3)
15070 TL=VAL(TVL$)*.001
15080 IF TR<>TL THEN PRINT"WRONG VALUE!TRY AGAIN" ELSE
PRINT"CORRECT!!":RETURN
15090 INPUT "INPUT THE TABLE VALUE AGAIN (last chance!)";TR
15100 CLS
15110 IF TR<>TL THEN PRINT"WRONG AGAIN!! THE CORRECT VALUE
IS ";TL:TR=TL ELSE PRINT"RIGHT ON!!"
15120 RETURN
20000 DEFSNG A-Z:ON ERROR GOTO 22000
20010 DIM TABLE$(19)
20020 FOR L=0 TO 19
20030  READ TABLE$(L)
20040 NEXT L
20050 DATA "997999999"
20051 DATA "950980990"
20052 DATA "878934959"
20053 DATA "811882917"
20054 DATA "754833875"
20055 DATA "707789834"
20056 DATA "666750798"
20057 DATA "632715765"
20058 DATA "602685735"
20059 DATA "576658708"
20060 DATA "553634684"
20061 DATA "532612661"
20062 DATA "514592641"
20063 DATA "497574623"
20064 DATA "482558606"
20065 DATA "468543590"
20066 DATA "456529575"
20067 DATA "444516561"
20068 DATA "433503549"
20069 DATA "423492537"
20080 RETURN
21980 '****************************************
21985 'ERROR TRAP
```

```
21990 '****************************************
22000 PRINT"ERROR";ERR;"IN LINE";ERL
22010 PRINT"CONSULT OPERATORS MANUAL":END
```

REFERENCES

1. TRS-80 Model 100 Owners/Operators Manual, Tandy Corporation, Fort Worth, TX, 1983.

2. Elementary Business Statistics: The Modern Approach, J. Freund, F. Williams, Prentice-Hall, Inc., England Cliffs, NJ, 1982.

3. OS 3101 Study Guide and Calculator Supplement, Peter W. Zehna, Naval Postgraduate School, Monterey, CA, 1982.

APPENDIX C

DESCRIPTION OF THE RADIO SHACK MODEL 100 PORTABLE MICROCOMPUTER

Dimensions: 11.8"(L) x 8.4"(D) x 2"(H)

Weight: 3 lbs., 13.5 oz.

Power Source: Battery (4 AA)
AC Adapter

Microprocessor: 80C84 (8 bit CPU) 2.4 MHz

RAM Capacity: 8, 24, or 32 k.
(version used was equipped with 24 k.)

Display Device: LCD screen -- 8 lines by 40 characters.

User Interface: Standard size typewriter keyboard.

Modem: Built in direct connect 300 baud modem.

Secondary Storage Capacity: No self-contained secondary storage device. Can be cable connected to either cassette tape or floppy disk devices.

LIST OF REFERENCES

1. Miller, William G., Microprocessors as an Adjunct to Statistics Instruction, p. 7, paper presented at the Annual Meeting of the American Educational Research Association, Toronto, Canada, April 1978.

2. Tubb, Gary W., Current Use of Computer in the Teaching of Statistics, pp. 18-20, paper presented at the Computer Science and Statistics Annual Symposium, 10th, Gaithersburg, Maryland, 14 April 1977.

3. Korbak, Michael Jr., Computer-Managed Instruction: Theory, Application, and Some Key Implementation Issues, pp. 16-21, Master's Thesis, Naval Postgraduate School, Monterey, California, March 1984.

4. Ibid.

5. Pitts, Gerald N. and Eggen, Maurice L., "CAI Course-ware Development Through Interactive Prompting", AEDS Twenty-first Annual Convention Proceedings, p. 242, May 1983.

6. Korbak, p. 16-18.

7. Ibid., p. 19.

8. Ibid., pp. 19-20.

9. Office of Naval Research Tech Memo No. 17, Current Research Development in Computer Assisted Instruction, p. 2, by D. N. Hansen.

10. Baughman, JoAnn, "Developing Instruction Material for the Computer," AEDS Twenty-first Annual Convention Proceedings, p. 10, May 1983.

11. Ibid., p. 10.

12. Ibid.

13. Ibid.

14. Computer-Based Education Research Laboratory Report No. 18, Myth and Reality: Essential Decisions in Computer-Based Instructional Design, by E. Call-Himwich and E.R. Steinberg, p. 17, 24 March 1978.

15. Baughman, p. 11.

16. Ibid.

17. Tubb, p. 19.

18. Hoban, C.F., "Focus on Learning", American Council on Education, Washington, DC, p. 27, 1942.

19. Stoddard, A.J., "Schools for Tomorrow", Fund for the Advancement of Education, p. 27, New York, New York, 1957.

20. Woefle, Dael, "Prophecy Fulfilled", Science, No. 3503, p. 503, 16 February 1962.

21. Suppes, P. and Mocken, E., "The Historical Path from Research and Development to Operational Use of CAI", Educational Technology, V. 18, pp. 9-112, 1978.

22. Hallworth, H.J. and Brebover, A., Computer Assisted Instruction in Schools: Achievements, Present Developments, and Projections for the Future, Vol. 1, p. 9, Alberta Department of Education, June 1980.

23. Kearsley, Greg P., Some "Facts" About CAI: Trends 1970-1976, pp. 3-4, Alberta University Division of Education Research Services, 1976.

24. Hallsworth, p. 14.

25. Ibid., p. 62.

26. Hall, Keith A., "Computer Based Education", Encyclopedia of Educational Research, V. 1, p. 355, 1982.

27. Ibid., p. 360.

28. Atkinson, Martha L., "Computer Assisted Instruction: Current State of the Art", Computers in the Schools, V. 1, p. 95, 1984.

29. Korbak, p. 8.

30. Douglass, V.G., "CAI in the Pocket", Software Review, V. 2, p. 73, 1983.

31. Atkinson, p. 91.

32. Stockburger, David W., *Teaching Statistics Using Microcomputer Simulations*, p. 16, paper presented at the Southwestern Psychological Association Meeting, Oklahoma City, Oklahoma, April 1980.

33. Ibid.

34. Kearsley, Greg P., "Some Conceptual Issues in Computer-Assisted Instruction", *Journal of Computer-Based Instruction*, V. 4, p. 9, August 1977.

35. Tubb, pp. 10-21.

36. Walker, Richard A. and Bergmann, Dann, "The Blossoming of Computer Aided Instruction", *PC Magazine*, p. 237, April 1983.

37. Sachdeva, Darshan, *The Role of the Computer in the Teaching of Business Statistics*, p. 113, paper presented at the Western Educational Computing Conference and Trade Show, California State University, Long Beach, November 1981.

38. Ryan, Thomas A. Jr., Joiner, Brian L., and Ryan, Barbara F., *MINITAB Student Handbook*, p. viii, Duxbury Press, PWS Publishers, 1976.

39. Stockburger, pp. 5-7.

40. Tubb, pp. 5-7.

41. Stockburger, pp. 4-8.

42. Atkinson, p. 91.

43. Office of Naval Research Systems Memo No. 12, *STATSIM Exercises in Statistics*, by D.B. Thomas, P.F. Merrill, and D.N. Hansen, p. 3, 15 April 1971.

44. Call-Himwich, pp. 16-30.

45. Hallworth, pp. 8-9.

46. Kearsley, Greg P., *Some Facts About CAI: Trends 1970-1976*, pp. 7-8, Aberta University Division of Educational Research Services, 1976.

47. Romano, A., *Computer Assisted Instruction*, p. 3.3.1, paper presented at the World Conference on Computer Education, Amsterdam, August 1970.

48. Kearsley, pp. 7-8.

49. Ibid.

50. Call-Himwich, pp. 19-22.

51. Baughman, p. 12.

52. Fawcette, Jim, "Finally Some Help For Documentation", Personal Computing, V. 7, pp. 141-143, December 1983.

53. Call-Himwich, pp. 19-21.

54. Ibid.

55. Roper, W.J., "Feedback in Computer Assisted Instruction," PLET, V. 14, pp. 43-48.

56. Ibid.

57. Ibid.

58. Call-Himwich, p. 22.

59. Baughman, p. 12.

60. Ibid.

61. Ibid.

62. Navy Personnel Research and Development Center TR-76-16, A Comparison of Three Combinatiions of Text and Graphics for Concept Learning, by William A. King, September 1975.

63. Baughman, p. 12.

64. Steinberg, pp. 10-32.

65. Hartman, Janet D., Creating Instructional Material for the Microcomputer,, paper presented at the Annual Convention Proceedings of AEDS, p. 21, Portland, Oregon, 9 May 1983.

66. Ibid.

67. Ibid.

68. Kearsley, p. 12.

69. Atkinson, p. 94.

70. Hartman, p. 128.

71. Pressman, R.S., *Software Engineering: A Practitioner's Approach*, pp. 147-176, McGraw-Hill, 1982.

72. Boehn, B.W., *Software Engineering Economics*, pp. 35-57, Prentice-Hall, 1981.

73. Pressman, pp. 289-321.

74: Ibid.

75. Ibid.

76. Ibid.

77. Kearsley, pp. 14-15.

78. Ibid.

BIBLIOGRAPHY

Advanced Research Project, Agency Report No. 25-DAHC-15-73-C-077, Implementation and Operation of Computer-Based Education, by Lynn A. Misselt and Others, 1980.

Air Force Office of Scientific Research AD/A006-424 Report 2992, *Optimizing the Efficiency of Computer-Aided Instruction*, by W. Feurzeig, G. Lukas, and N. Benhaim, January 1975.

Anderson, G.E., "Using a Microcomputer to Help Teach Beginning Statistics", *AEDS Monitor*, v. 21, pp. 37-43, September/October 1982.

Appelbe, William, and Volper, Dennis, *A Portable Software Environment for Microcomputers*, paper presented at the Computer Science and Statistics 12th Annual Symposium on the Interface, Ontario, Canada, 10-11 May 1979.

Bellissant, C., *Teaching and Learning Languages*, paper presented at the World Conference on Computer Education, Amsterdam, 1970.

Bickford, Robert, "Micros Still Can't Provide Full CAI-An Opposing Viewpoint", *Infoworld*, v. 5, no. 43, 24 October 1983.

Caffarella, E.P., "Using an Interactive Computer System to Teach Statistical Methods in Education", *Journal of Educational Technology Systems*, v. 11(1), 1982-1983.

Charp, Sylvia, *Computer Technology in Education: How to Make It Viable*, paper presented at the World Conference on Computer Education, Amsterdam, 24-28 August 1970.

Countermine, Terry, and Singh, Jane M., "A CMI Support System for Large Group Individualized Instruction", *Journal of Computer-Based Instruction*, v. 4, no. 1, pp. 17-21, August 1977.

Dean, Peter M., "Why CBI? An Examination of the Case for Computer-Based Instruction", *Journal of Computer-Based Instruction*, v. 4, no. 1, pp. 1-7, August 1977.

Dixon, Paul N., and Judd, Wilson A., "A Comparison of CMI and Lecture Mode for Teaching Basic Statistics", Journal of Computer-Based Instruction, v. 4, no. 1, pp. 22-25, August 1977.

Dwass, Meyer, "The Use of the Computer in Teaching Statistics", Supplemental Advanced Applied Probability, v. 7, pp. 12-16, 1975.

Emond, W.J., "Some Benefits of Micro-Computers in Teaching Statistics", Computers and Education, v. 6, pp. 51-54, 1982.

Hartley, J.R., and Abbatt, F.R., The Design and Use of Computer-Based Programs in Applied Statistics, paper presented at the International Conference on Frontiers in Education, Leeds, UK, 15-19 July 1974.

Kearsley, Greg P., Computer-Based Education Around the World: Perspective, Problems, and Prospects, Division of Educational Services, University of Alberta, Canada, 1975.

Knief, L.M., and Cunningham, G.K., "Effects of Tutorial CBI on Performance in Statistics", AEDS Journal, Winter, 1976.

Naval Postgraduate School Report NPS55LW73061A, Random Number Generation Package LL Random, by G.P. Learmonth and P.A.W. Lewis, pp. 1-25, June 1973.

Levy, C.M., "Microcomputer-Based Tutorials for Teaching Statistics", Behavior Research Methods and Instrumentation, v. 15(2), pp. 127-129, 1983.

Masterson, Fred A., "Languages for Students", Byte, v. 6, June 1984.

Moonen, J., "The Teaching of Statistics and CAL", Computer-Assisted Learning, v. 1, pp. 27-36, 1980.

Moskova, T.T., and Moskov, A.T., An Optimal Strategy on Computer Education, paper presented at the World Conference on Computer Education, Amsterdam, 1970.

Nielen, G.C., and Scheepmaker, B., Information Systems: Approach for a Curriculum, paper presented at the World Conference on Computer Education, Amsterdam, 1970.

Office of Naval Research Contract NONR-624(18), Computer-Assisted Instruction in Statistics, by W.W. Cooley, 1969.

Office of Naval Research, Report No. 11, *Development and Implementation of a Computer-Managed Instruction System in Graduate Training*, by Nancy K. Hagerty, 30 June 1970.

Office of Naval Research NONR-624(18), *Studies Related to Computer-Assisted Instruction*, by Robert Glaser, May 1969.

ONR Conference on CAI Languages, Cambridge, Massachusetts, 2-3 March 1966.

Pask, Gordon, *Fundamental Aspect of Computer Technology*, paper presented at the World Conference on Computer Education, Amsterdam, 1970.

Pavlick, Frank M., "The Attitudinal Effect of Using the Computer in an Elementary Statistics Course", *International Journal of Math, Education, Science and Technology*, v. 6, pp. 353-360, 1975.

Pukkila, Tarmo, and Puntanon, Simo, "The Computer as an Aid in Teaching Basic Statistics Courses at University Level", *Computers in Education*, 1981.

Pukkila, Tampore T., *The Utilization of the Computer in the Teaching of Statistics at Elementary Level*, paper presented at the Compstat 3rd Symposium, 1978.

Quasney, James S., *An Approach to Statistics Through the Use of a Programmable Calculator*, paper presented on 7 March 1975.

Schaffner, Paul E., "A Self-Paced CAI Package for Selecting Statistical Methods", *Behavior Research Methods and Instrumentation*, v. 15(2), pp. 130-134, 1983.

System Development Corporation TM2914-100, *Computer-Based Instruction in Statistical Inference*, by J. Rosenbaum, and others, 30 October 1967.

Sytsma, Sidney P., *Use of the Personal Computer as an Aid in Class Teaching Tool in Introductory Business Statistics*, paper presented at the Association for Educational Data Systems 17th Annual Conference, 14-18 May 1970.

Ting, T.C., *A Database Oriented Computer Assisted Instruction System*, paper presented at the ONLINE 72 Conference, Blacksburg, Virginia, 1972.

TRACOR Science and Systems Division SD-67-016-U, *Toward the Development of a Systematic Functional Methodology for Computer Assisted Instruction*, July 1967.

Vollebergh, J.J.A., The Challenge of Computer Education, paper presented at the World Conference on Computer Education, Amsterdam, 1970.

Wegman, E.J., and Gere, B.H., "Some Thoughts on Computers and Introductory Statistics", International Journal of Math, Education, Science and Technology, v. 3, pp. 211-221, 1972.

INITIAL DISTRIBUTION LIST

No. Copies

1. Defense Technical Information Center 2
Cameron Station
Alexandria, Virginia 22304-6145

2. Library, Code 0142 2
Naval Postgraduate School
Monterey, California 93943-5100

3. Curricular Office, Code 37 1
Computer Technology
Naval Postgraduate School
Monterey, California 93943-5100

4. Assistant Professor Tung Bui, Code 54Bd 5
Department of Administrative Sciences
Naval Postgraduate School
Monterey, California 93943-5100

5. LTCOL F. Marchman Perry, USA, Code 55Pj 1
Department of Operations Research
Naval Postgraduate School
Monterey, California 93943-5100

6. Associate Professor J. N. Eagle, II, Code 55Er 1
Department of Operations Research
Naval Postgraduate School
Monterey, California 93943-5100

7. LTCOL Richard L. Forney, USAF, Code 56Fk 1
National Security Affairs/Intelligence
Naval Postgraduate School
Monterey, California 93943-5100

8. LT Randall Henderson, USN 1
NIPSTRAFAC
NAS, Key West
Key West, Florida 33040

214329

Thesis
H4397
c.1

Henderson
Computer aided instruction design issues: the development of a portable microcomputer based CAI for statistics instruction.

~~4 MAR 87~~ 33344

214329

Thesis
H4397
c. 1

Henderson
Computer aided instruction design issues: the development of a portable microcomputer based CAI for statistics instruction.

AMERICAN BINDERIES

thesH4397
Computer aided instruction design issues
3 2768 000 64923 0
DUDLEY KNOX LIBRARY